Panasonic®

PROGRAMMABLE CONTROLLERS

FP0R

User's Manual

BEFORE BEGINNING

Liability and Copyright for the Hardware

This manual and everything described in it are copyrighted. You may not copy this manual, in whole or part, without written consent of Panasonic Electric Works Europe AG (PEWEU).

PEWEU pursues a policy of continuous improvement of the design and performance of its products. Therefore we reserve the right to change the manual/product without notice. In no event will PEWEU be liable for direct, special, incidental, or consequential damage resulting from any defect in the product or its documentation, even if advised of the possibility of such damages.

We invite your comments on this manual. Please e-mail us at:

tech-doc@eu.pewg.panasonic.com.

Please direct support matters and technical questions to your local Panasonic representative.

LIMITED WARRANTY

If physical defects caused by distribution are found, PEWEU will replace/repair the product free of charge. Exceptions include:

- When physical defects are due to different usage/treatment of the product other than described in the manual.
- When physical defects are due to defective equipment other than the distributed product.
- When physical defects are due to modifications/repairs by someone other than PEWEU.
- When physical defects are due to natural disasters.

Important Symbols

One or more of the following symbols may be used in this documentation:

DANGER!

The warning triangle indicates especially important safety instructions. If they are not adhered to, the results could be fatal or critical injury.

◆CAUTION

Indicates that you should proceed with caution. Failure to do so may result in injury or significant damage to instruments or their contents, e.g. data.

◆NOTE

Contains important additional information.

◆EXAMPLE

Contains an illustrative example of the previous text section.

◆Procedure

Indicates that a step-by-step procedure follows.

◆REFERENCE

Indicates where you can find additional information on the subject at hand.

Scope of This Manual

The FP0R User's Manual includes:

- specifications for the CPU types and expansion units of the FP0R
- installation, wiring, and maintenance instructions
- general programming information
- troubleshooting information
- an appendix with:
  - technical specifications
  - I/O allocation tables
  - memory area tables
  - system registers
  - unit dimensions

◆REFERENCE

Please refer to the FP Series Programming Manual, or to the online help of FPWIN Pro or FPWIN GR for information on:

- system instructions
- special internal relays
- data registers
- system variables (FPWIN Pro only)
  - **programming examples**

For documentation on a particular unit used with the FP0R, please refer to the hardware manual for that unit.

All manuals can be downloaded from the **Panasonic Web site** (http://www.panasonic-electric-works.com).

Programming Conventions

The programming examples in this manual are designed for FPWIN Pro. For FPWIN GR examples, please refer to: FP0R User's Manual ARCT1F475E

Most of the sample programs were written in Ladder Diagram. In FPWIN Pro, you can also program in Structured Text, Function Block Diagram, Instruction List, and Sequential Function Chart. For examples in other programming languages, please refer to the FPWIN Pro Online Help and the Programming Manual.

The abbreviations used in the examples signify the following:

- POU: Program Organization Unit
- DUT: Data Unit Type
- GVL: Global Variable List

These and other terms are explained in the FPWIN Pro Online Help and Programming Manual.

To illustrate the use of positioning instructions, the chapter on high-speed counters and pulse output contains numerous examples. Some of the sample programs can be opened directly in FPWIN Pro. The FPWIN Pro projects in LD and ST code can be downloaded from the Panasonic Web site (http://www.panasonic-electric-works.com/peweu/en/html/22164.php).

Table of Contents

1. **Safety Measures 1**

1.1 Safety Measures 2

2. **Overview 5**

2.1 Features 6

2.2 Unit Types 8

2.2.1 CPU 8

2.2.2 FP0 I/O Expansion Units 9

2.2.3 FP0 Intelligent Units 9

2.2.4 FP Series Link Units 10

2.2.5 Power Supply Unit 10

2.2.6 Accessories 11

2.3 Restrictions on Unit Combinations 12

2.4 Programming Tools 13

2.5 FP0 Program Compatibility 14

3. **CPU Types 17**

3.1 Parts and Functions, CPU 18

3.2 Input Specifications, CPU 22

3.3 Output Specifications, CPU 24

3.4 Terminal Layout 27

3.4.1 C10 CPU 27

3.4.2 C14 CPU 28

3.4.3 C16 CPU 29
3.4.4 C32 CPU 30
3.5 Backup and Clock/Calendar Functions 31
3.5.1 Backup Function 32
3.5.2 Clock/Calendar Function 33
3.5.2.1 Memory Area for Clock/Calendar Function 33
3.5.2.2 Settings for Clock/Calendar Function 34
3.5.2.3 Sample Program for Fixed Schedule and Automatic Start 35
3.5.2.4 Sample Program for 30-Second Compensation 35

4. Expansion 39

4.1 Expansion Method 40
4.2 Parts and Functions, Expansion Units 41
4.3 Input Specifications, Expansion Units 43
4.4 Output Specifications, Expansion Units 44
4.5 Terminal Layout 46

5. I/O Allocation 49

5.1 General 50
5.2 CPU 51
5.3 FP0 Expansion Units 52

6. Installation and Wiring 55

6.1 Installation 56
6.1.1 Installation Environment and Space 56
6.1.2 Using DIN Rails 58
6.1.3 Using Optional Mounting Plates 59

6.1.3.1 Slim Type Mounting Plate 59
6.1.3.2 Flat Type Mounting Plate 60
6.2 Connecting FP0 Expansion Units 63
6.3 Safety Instructions for Wiring 64
6.4 Wiring the Power Supply 66
6.4.1 Grounding 68
6.5 Input and Output Wiring 70
6.5.1 Input Wiring 70
6.5.2 Output Wiring 73
6.5.2.1 Protective Circuit for Inductive Loads 73
6.5.2.2 Protective Circuit for Capacitive Loads 74
6.6 Wiring the MIL Connector 75
6.7 Wiring the Terminal Block 76
6.8 Wiring the COM Port 78

7. **Communication 81**

7.1 Communication Modes 82
7.1.1 Terminology in FPWIN Pro and FPWIN GR 82
7.1.2 MEWTOCOL-COM Master/Slave 82
7.1.3 Program Controlled Communication 83
7.1.4 PLC Link 84
7.1.5 Modbus RTU Master/Slave 85
7.2 Ports: Names and Principle Applications 86
7.2.1 TOOL Port 86
7.2.2 COM Port 87
7.2.3 USB Port 87
7.2.3.1 Installing the USB Driver 89
7.2.3.2 Communication with the Programming Tool 91
7.2.3.3 Reinstalling the USB Driver 91

7.3 Communication Specifications 93
7.4 Communication Parameters 94
7.4.1 Setting System Registers in PROG Mode 94
7.4.2 Changing Communication Mode in RUN Mode 96
7.5 MEWTOCOL-COM 97
7.5.1 Operation Outline for MEWTOCOL-COM Slave 99
7.5.2 Command and Response Format 100
7.5.3 Commands 102
7.5.4 Setting Communication Parameters 103
7.5.4.1 FP0 Compatibility Mode 103
7.5.5 1:1 Slave Communication 104
7.5.5.1 1:1 Communication with a Computer 105
7.5.5.2 1:1 Communication with GT Series Programmable Displays 106
7.5.6 1:N Slave Communication 107
7.5.7 Sample Program for Master Communication 108
7.6 Program Controlled Communication 110
7.6.1 Setting Communication Parameters 111
7.6.1.1 FP0 Compatibility Mode 112
7.6.2 Sending Data to External Devices 113
7.6.3 Receiving Data from External Devices 117
7.6.4 Data Format 120
7.6.5 Flag Operation 121
7.6.5.1 Start Code: No-STX; End Code: CR 123
7.6.5.2 Start Code: STX; End Code: ETX 124
7.6.6 1:1 Communication 126
7.6.6.1 1:1 Communication with Micro-Imagechecker 127
7.6.7 1:N Communication 133
7.6.8 Setting Communication Parameters 134
7.7 PLC Link 135
7.7.1 Setting Communication Parameters 137
7.7.2 Link Area Allocation 138
7.7.2.1 Example for PLC link 0 139
7.7.2.2 Example for PLC link 1 140

7.7.2.3 Partial Use of Link Areas 142
7.7.2.4 Precautions for Allocating Link Areas 143
7.7.3 Setting the Highest Station Number for a PLC Link 144
7.7.4 PLC Link 0 and 1 Allocation Setting 144
7.7.5 Monitoring 145
7.7.6 PLC Link Response Time 146
7.7.6.1 Reducing Transmission Cycle Times 149
7.7.6.2 Error Detection Time for Transmission Assurance Relays 150
7.8 Modbus RTU Communication 152
7.8.1 Setting Communication Parameters 155
7.8.2 Sample Program for Master Communication 155

8. **High-Speed Counter and Pulse Output 157**

8.1 Overview 158
8.2 Function Specifications and Restrictions 160
8.2.1 High-Speed Counter Function 160
8.2.2 Pulse Output Function 161
8.2.3 PWM Output Function 162
8.2.4 Maximum Counting Speed and Output Frequency 163
8.3 High-Speed Counter Function 166
8.3.1 Count Input Modes 166
8.3.2 Minimum Input Pulse Width 168
8.3.3 I/O Allocation 168
8.3.4 Instructions and System Variables 169
8.3.4.1 Writing the High-Speed Counter Control Code 170
8.3.4.2 Writing and Reading the Elapsed Value for the High-Speed Counter 173
8.3.4.3 F166_HighSpeedCounter_Set, Target Value Match ON 173
8.3.4.4 F167_HighSpeedCounter_Reset, Target Value Match OFF 174
8.3.4.5 F178_HighSpeedCounter_Measure, Input Pulse Measurement 175
8.3.5 Sample Programs 175
8.3.5.1 Positioning Operations with a Single-Speed Inverter 176

8.3.5.2 Positioning Operations with a Double-Speed Inverter 177

8.4 Pulse Output Function 179

8.4.1 Pulse Output Methods and Position Control Modes 179

8.4.2 I/O Allocation 182

8.4.3 Instructions and System Variables 183

8.4.3.1 Writing the Pulse Output Control Code 185

8.4.3.2 Writing and Reading the Elapsed Value of the Pulse Output 188

8.4.3.3 F166_PulseOutput_Set, Target Value Match ON 189

8.4.3.4 F167_PulseOutput_Reset, Target Value Match OFF 190

8.4.3.5 F171_PulseOutput_Trapezoidal, Trapezoidal Control 191

8.4.3.6 F171_PulseOutput_Jog_Positioning, JOG Operation and Positioning 192

8.4.3.7 F172_PulseOutput_Jog, JOG Operation 194

8.4.3.8 F174_PulseOutput_DataTable, Data Table Control 195

8.4.3.9 F175_PulseOutput_Linear, Linear Interpolation 196

8.4.3.10 F177_PulseOutput_Home, Home Return 196

8.5 PWM Output Function 198

8.5.1 PWM Output Function 198

9. **Security Functions 201**

9.1 Security Function Types 202

9.2 Security Settings in FPWIN Pro 203

9.2.1 Upload Protection 203

9.2.2 PLC Protection (Password Protection) 204

9.3 FP Memory Loader 205

9.3.1 Upload Protection 205

9.3.2 Download Protection 206

10. **Other Functions 209**

10.1 F-ROM Backup (P13_EPWT) 210

10.2 Sampling Trace 211
10.3 Input Time Constants 212

11. **Troubleshooting 213**

11.1 LED Display for Status Condition 214
11.2 Operation on Error 215
11.3 ERROR/ALARM LED is Flashing 216
11.4 ERROR/ALARM LED is ON 217
11.5 All LEDs are OFF 218
11.6 Diagnosing Output Malfunction 219
11.7 Password Protection Error Message 220
11.8 PROG Mode does not Change to RUN 221

12. **Appendix 223**

12.1 Specifications 224
12.1.1 General Specifications 224
12.1.2 Performance Specifications 225
12.1.3 Communication Specifications 227
12.1.4 Current Consumption 228
12.1.5 I/O Allocation 229
12.2 Dimensions 231
12.2.1 C10/C14 CPU (Terminal Block) 231
12.2.2 C16 CPU (MIL Connector) 233
12.2.3 C32 CPU (MIL Connector) 235
12.2.4 Power Supply Unit 236
12.2.5 I/O Link Unit 236
12.2.6 Using DIN Rails 237

12.3 Relays and Memory Areas for FP0R....238

12.4 System Registers....241

12.4.1 Precautions When Setting System Registers....241

12.4.2 Types of System Registers....241

12.4.3 Checking and Changing System Registers....242

12.4.4 Table of System Registers....243

12.5 Error Codes....249

12.5.1 Error Codes E1 to E8....249

12.5.2 Self-Diagnostic Error Codes....250

12.5.3 MEWTOCOL-COM Error Codes....250

12.6 MEWTOCOL-COM Communication Commands....252

12.7 Data Types....253

12.8 Hexadecimal/Binary/BCD....254

12.9 ASCII Codes....255

13. **Index....257**

Chapter 1

Safety Measures

1.1 Safety Measures

Operating environment

After installing the unit, make sure to use it within the range of the general specifications:

- Ambient temperature: 0°C–+55°C
- Ambient humidity: 10%–95% RH (at 25°C non-condensing)
- Pollution level: 2
- Do not use the unit in the following environments:
  - Direct sunlight
  - Sudden temperature changes causing condensation
  - Inflammable or corrosive gases
  - Excessive airborne dust, metal particles or salts
  - Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda
  - Direct vibration, shock or direct drop of water
  - Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges. Maintain at least 100mm of space between these devices and the unit.

Static electricity

- Before touching the unit or equipment, always touch some grounded metal to discharge any static electricity you may have generated (especially in dry locations). The discharge of static electricity can damage parts and equipment.

Protection of power supply

- Use a twisted power supply wire.
- Isolate the wiring systems to the CPU, input/output devices, and mechanical power apparatus.
- An insulated power supply with an internal protective circuit should be used (FP0-PSA2 or FP-PS24-050). The power supply for the CPU is a non-insulated circuit, so if an incorrect voltage is directly applied, the internal circuit may be damaged or destroyed.
- If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.
- Be sure to supply power to a CPU and an expansion unit from the same power supply, and turn the power on and off simultaneously for both.

Power supply sequence

- Make sure the power supply of the CPU turns off before the power supply for input

and output. If the power supply for input and output is turned off first, the CPU will detect the input fluctuations and may begin an unexpected operation.

Before turning on the power

When turning on the power for the first time, be sure to take the precautions given below.

- During installation, check that there are no scraps of wiring, particularly conductive fragments, adhering to the unit.
- Verify that the power supply wiring, I/O wiring, and power supply voltage are all correct.
- Sufficiently tighten the installation and terminal screws.
- Set the mode selector to PROG mode.

Before entering a program

Be sure to clear any existing program before entering a new program.

◆Procedure

1. **Online → Online Mode**
2. **Online → Clear Program and Reset System Register**
3. **Choose [OK] in the confirmation dialog box**

Request concerning program storage

To prevent the accidental loss of programs, the user should consider the following measures:

- Backing up programs. To avoid accidentally losing programs, destroying files, or overwriting the contents of a file, documents should be printed out and then saved.
- Specifying the password carefully. The password setting is designed to avoid programs being accidentally overwritten. If the password is forgotten, however, it will be impossible to overwrite the program even if you want to. Also, if a password is forcibly bypassed, the program is deleted. Therefore, please note the password in a safe location.

Chapter 2

Overview

2.1 Features

The FP0R is an ultra compact PLC (programmable logic controller) with high-speed processing capabilities and a large memory capacity. The controller uses the comprehensive FP instruction set and is programmed with FPWIN Pro or FPWIN GR. With FPWIN Pro, programming according to IEC 61131-3 is possible.

USB 2.0 TOOL port (see page 87)

The TOOL port supports USB2.0 full speed and enables ultra high-speed communication with programming tools. Since large programs with up to 32k steps can now be downloaded in as fast as 5s, the USB port enables more efficient program development.

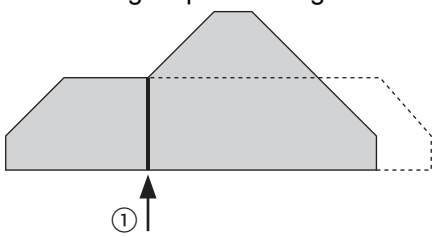
Separate large-capacity comment memory

The unit's comment memory area is separate from the program area, and can store I/O comments for 100 000 points. Program management and maintenance is easy. Thanks to the separate comment area, programs can now be developed without concern for comment memory capacity.

Positioning control using high-speed counter and pulse output (see page 160)

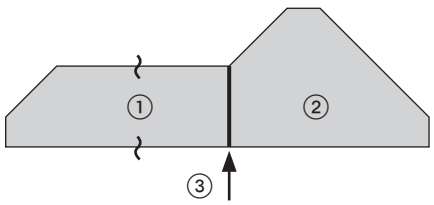
A high-speed counter and a pulse output function are provided as standard features.

- Target speed change



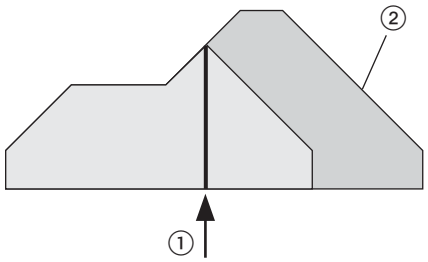
① Target speed change

- JOG operation



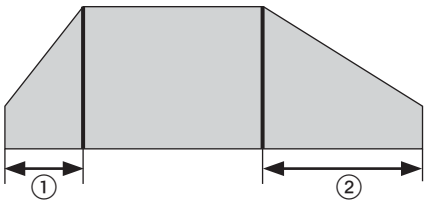
① JOG operation
② Total movement amount [pulses]
③ Position control trigger input

- Decelerated stop



① Trigger for decelerated stop
② Total movement amount [pulses]

- Individual setting for acceleration/deceleration time



① Acceleration time
② Deceleration time

Additional unit with battery free backup function (F32 type) (see page 32)

The F32 type offers a battery-free automatic backup function for all operation memories (internal relays, data registers, timers/counters). Maintainability has been significantly improved, as there is no need to change a battery.

Full range of communication functions (see page 81)

- PLC Link (supports MEWNET-W0)
- MEWTOCOL-COM Master/Slave
- MODBUS RTU Master/Slave
- Program controlled communication via TOOL or COM port (RS232C)

Extended online editing functions

Additional functions now ensure that programs can be corrected without stopping the system. The online edit mode is no longer limited to 512 steps. Instead, entire programs can be downloaded to the program memory during RUN mode. Project information is written to the comment memory. Please refer to the FPWIN Pro online help for detailed information.

Enhanced security (see page 201)

The FP0R supports 8-digit passwords (alphanumeric), and offers an upload protection function as well as security functions for the FP Memory Loader.

FP0 compatibility (see page 14)

The FP0 compatibility mode enables programs that have been used on an existing FP0 to be activated on the FP0R with no further modifications. Also, since both units have an identical shape and terminal layout, there is no need to check the installation space or change the wiring.

2.2 Unit Types

The following units are available for the FP0R:

2.2.1 CPU

16k types (program capacity: 16k steps)

Type	Number of I/O points [1]	Power supply	Input	Output	Connection	COM port[2]	Product no.
C10	10 (6/4)	24V DC	24V DC ±COM terminal	Relay: 2A	Terminal block	—	AFP0RC10RS
						X	AFP0RC10CRS
C14	14 (8/6)					—	AFP0RC14RS
						X	AFP0RC14CRS
C16	16 (8/8)			Transistor (NPN): 0.2A	MIL connector	—	AFP0RC16T
				Transistor (PNP): 0.2A		—	AFP0RC16P
				Transistor (NPN): 0.2A		X	AFP0RC16CT
				Transistor (PNP): 0.2A		X	AFP0RC16CP

[1] Total number (input points/output points)

[2] RS232 C interface (X: Available)

32k types (program capacity: 32k steps)

Type	Number of I/O points [1]	Power supply	Input	Output	Connection	COM port[2]	Product no.
C32	32 (16/16)	24V DC	24V DC ±COM terminal	Transistor (NPN): 0.2A	MIL connector	—	AFP0RC32T
				Transistor (PNP): 0.2A		—	AFP0RC32P
				Transistor (NPN): 0.2A		X	AFP0RC32CT
				Transistor (PNP): 0.2A		X	AFP0RC32CP
T32				Transistor (NPN): 0.2A		X	AFP0RT32CT
				Transistor (PNP): 0.2A		X	AFP0RT32CP
F32				Transistor (NPN): 0.2A		X	AFP0RF32CT
				Transistor (PNP): 0.2A		X	AFP0RF32CP

[1] Total number (input points/output points)

[2] RS232 C interface (X: Available)

2.2.2 FP0 I/O Expansion Units

Type	Number of I/O points	Power supply	Input	Output	Connection	Product no.
Expansion unit E8	8 (8/–)	–	24V DC ±COM terminal	–	MIL connector	FP0-E8X
	8 (4/4)	24V DC	24V DC ±COM terminal	Relay: 2A	Terminal block	FP0-E8RS
	8 (–/8)	24V DC	–	Relay: 2A	Terminal block	FP0-E8YRS
	8 (–/8)	–	–	Transistor (NPN): 0.1A	MIL connector	FP0-E8YT
	8 (–/8)	–	–	Transistor (PNP): 0.1A	MIL connector	FP0-E8YP
Expansion unit E16	16 (16/–)	–	24V DC ±COM terminal	–	MIL connector	FP0-E16X
	16 (8/8)	24V DC	24V DC ±COM terminal	Relay: 2A	Terminal block	FP0-E16RS
	16 (8/8)	–	24V DC ±COM terminal	Transistor: (NPN) 0.1A	MIL connector	FP0-E16T
	16 (8/8)	–	24V DC ±COM terminal	Transistor: (PNP) 0.1A	MIL connector	FP0-E16P
	16 (–/16)	–	–	Transistor: (NPN) 0.1A	MIL connector	FP0-E16YT
	16 (–/16)	–	–	Transistor: (PNP) 0.1A	MIL connector	FP0-E16YP
Expansion unit E32	32 (16/16)	–	24V DC ±COM terminal	Transistor: (NPN) 0.1A	MIL connector	FP0-E32T
	32 (16/16)	–	24V DC ±COM terminal	Transistor: (PNP) 0.1A	MIL connector	FP0-E32P

2.2.3 FP0 Intelligent Units

Type	Specifications	Product no.	Manual
FP0 thermocouple unit	Thermocouple types: K, J, T, R (Resolution 0.1°C)	FP0-TC4	ARCT1F366
	Thermocouple types: K, J, T, R (Resolution 0.1°C)	FP0-TC8	
FP0 analog I/O unit	No. of input channels: 2 Input range (Resolution 1/4000): • Voltage: 0–5V, -10–+10V • Current: 0–20mA	FP0-A21	ARCT1F390
	No. of output channels: 1 Output range (Resolution 1/4000): • Voltage: -10–+10V • Current: 0–20mA		

Type	Specifications	Product no.	Manual
FP0 A/D conversion unit	No. of input channels: 8 Input range (Resolution 1/4000): • Voltage: 0–5V, -10–+10V, -100–100mV • Current: 0–20mA	FP0-A80	ARCT1F321
FP0 D/A conversion unit	No. of output channels: 4 Output range (Resolution 1/4000): • Voltage: -10–+10V • Current: 4–20mA	FP0-A04V	ARCT1F382
		FP0-A04I	
FP0 RTD unit	Pt100, Pt1000, Ni1000 Resolution: 0.1°C/0.01°C (depending on switch setting)	FP0-RTD6	ARCT1F445

2.2.4 FP Series Link Units

Type	Specifications	Power supply	Product no.	Manual
FP0 I/O link unit	Designed to make the FP0 function as a MEWNET-F slave unit (remote I/O system).	24V DC	FP0-IOL	This manual
C-NET adapter S2 (for FP0 side)	RS485 adapter for connecting PLC and host via C-NET using MEWTOCOL-COM. Supplied with a 30cm FP0 TOOL port cable. A power supply is not required.	–	–	ARCT1F96
C-NET adapter (for computer side)	RS485 adapter for connecting PLC and host via C-NET using MEWTOCOL-COM.	100–240V AC	–	
		24V DC	–	
FP Web-Server 2	Designed to connect FP series PLCs to the Ethernet, to send e-mails, and present PLC data as HTML pages.	–	FP-WEB2	ARCT1F446

2.2.5 Power Supply Unit

Product name	Specifications	Product no.
FP0 power supply	Input voltage: 85–265V AC Max. output current: 0.7A (24V DC)	FP0-PSA2
FP power supply	Input voltage: 85–265V AC Max. output current: 2.1A (24V DC)	FP-PS24-050E

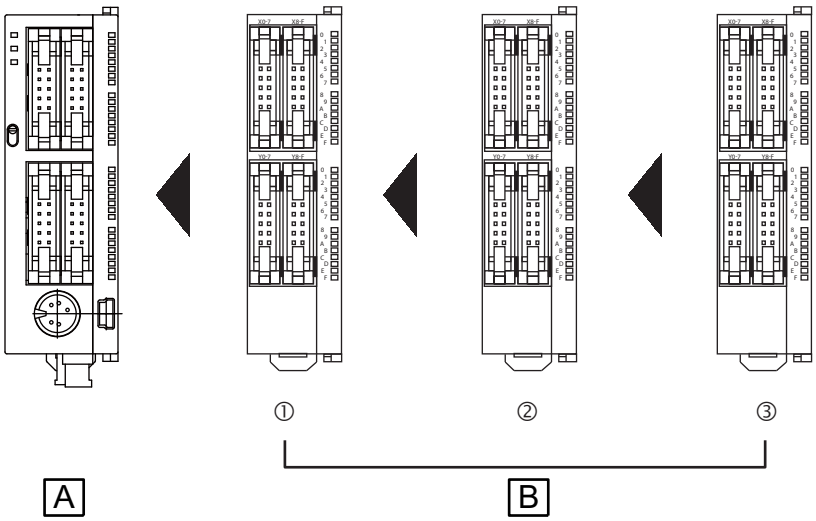
2.2.6 Accessories

Name	Description		Product no.
I/O cable	10-pin MIL wire-press socket on one side, 2 pieces (blue, white, or multi-colored)	Cable length: 1m	AFP0521D AFP0521BLUED AFP0521COLD
		Cable length: 3m	AFP0523D AFP0523BLUED
FP0 power supply cable	Maintenance part (packed with FP0 units)	Cable length: 1m	AFP0581
FP0R/FPΣ power supply cable	Maintenance part (packed with CPU)	Cable length: 1m	AFPG805
Phoenix connector (2 pcs)	Terminal block socket; maintenance parts (packed with relay output type)		AFP0802
Connector set	40-pin MIL wire-press socket; maintenance parts (packed with I/O expansion unit); 2 pieces		AFP2801
MIL connector (2 pcs)	10-pin MIL wire-press socket; maintenance parts (packed with transistor output type)		AFP0807
FP0 slim 30 type mounting plate (10 pcs)	For vertical mounting of the FP0R C32/T32/F32 CPU		AFP0811
FP0 slim type mounting plate (10 pcs)	For vertical mounting of FP0 expansion units		AFP0803
Flat type mounting plate (10 pcs)	For horizontal mounting of the CPU		AFP0804
FP Memory Loader	For reading/writing programs from/to PLC	Data clear type	AFP8670
		Data hold type	AFP8671

2.3 Restrictions on Unit Combinations

By adding expansion units, the number of I/O points can be increased. However, the maximum number of expansion units per CPU is limited.

A maximum of three expansion units can be connected on the right side of the FP0R CPU, these expansion units being either I/O expansion units or intelligent units. A combination of relay output types and transistor output types is also possible.



A	FP0R CPU
B	Maximum expansion: 3 units
①	Expansion unit 1
②	Expansion unit 2
③	Expansion unit 3

Maximum number of I/O points

CPU type	CPU	Using expansion units of same output type	Using transistor type expansion units
C10	10	58	106
C14	14	62	110
C16	16	112	112
C32	32	128	128
T32			
F32			

◆**NOTE**

- **Install the FP0 thermocouple unit to the right of other expansion units. If it is installed on the left side, overall precision will deteriorate. For details, refer to the FP0 thermocouple unit manual.**
- **Install the FP0 RTD unit to the right of the other expansion units.**

2.4 Programming Tools

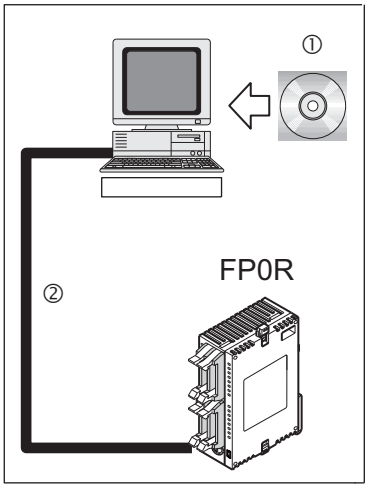
① Programming software

You can use the following programming software to program the FP0R:

- FPWIN Pro Version 6 or later
- FPWIN GR Version 2 or later

FP Memory Loader (AFP8670/AFP8671) to transfer programs and system registers can also be used.

You can connect your PC to the FP0R with an RS232C programming cable or via the USB port.



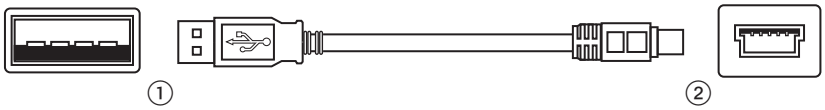
② PC connection cable:

RS232C programming cable

Connector	Description	Product no.
9-pin Sub-D to 5-pin Mini-DIN (round)	Programming cable for FP and GT series	AFC8513D

USB cable

Cable type	Length	Product no.
USB 2.0 (or 1.1) AB type	2m	CABMINIUSB5D



①	A type (male), PC side
②	5-pin Mini-B type (male), PLC side

Instead of Panasonic's USB cable any commercial USB cable meeting the above specifications may be used. The maximum permissible cable length is 5m.

2.5 FP0 Program Compatibility

For programs used on the existing FP0 to be used on the FP0R, they must either:

1. conform to FP0R specifications, or
2. be executed in FP0 compatibility mode

Using programs that conform to FP0R specifications

This allows you to make maximum use of FP0R performance and functions. However, the following modifications to the FP0 program are required:

1. Change the PLC type from FP0 to FP0R.
2. Since the system registers will be initialized when the PLC type is changed, reconfigure the system registers if necessary.
3. Modify the programs according to the FP0R specifications if necessary.

Executing programs in FP0 compatibility mode

The FP0 compatibility mode allows you to use existing FP0 programs as they stand. For items such as program capacity, operation memory and available instructions, the same specifications apply as for the FP0.

To enter the FP0 compatibility mode, use your programming tool to download the FP0 programs. A confirmation message will appear, and the mode will automatically change to the FP0 compatibility mode. The FP0 program may either have been uploaded from an FP0 or it may have been created on an FP0R in FP0 mode (PLC type is FP0).

The FP0 compatibility mode is supported by FPWIN Pro V6.10 or higher, and FPWIN GR V2.80 or higher.

For an FP0 program to be able to run in FP0 compatibility mode, the PLC types (C10, C14, C16, C32, and T32) must match exactly. FP0 compatibility mode is not available for the F32 type FP0R.

In most respects, the FP0 programs do not need to be modified to be executable in FP0 compatibility mode. Please note, however, the following differences between the specifications, and change the programs as necessary:

1. P13_EPWT, EEPROM write instruction

   The execution times for this instruction will vary, depending on the number of write blocks.

No. of write blocks (words)	FP0 [ms]	FP0 compatibility mode [ms]
1 (64)	≈5	≈100
2 (128)	≈10	≈100
4 (256)	≈20	≈100
8 (512)	≈40	≈100
16 (1024)	≈80	≈100
32 (2048)	≈160	≈100
33 (2112)	≈165	≈200
41 (2624)	≈205	≈200

No. of write blocks (words)	FP0 [ms]	FP0 compatibility mode [ms]
64 (4096)	≈320	≈200
96 (6144)	≈480	≈300
256 (16320)	≈800	≈800

2. F170_PulseOutput_PWM, PWM output instruction

   The frequency settings differ. In particular, the setting for the low-frequency band cannot be defined.

	FP0		FP0 compatibility mode	
K	Frequency [Hz]	Period [ms]	Frequency [Hz]	Period [ms]
8	0.15	6666.7	Cannot be specified.	
7	0.3	3333.3		
6	0.6	1666.7		
5	1.2	833.3		
4	2.4	416.7		
3	4.8	208.3	6	166.7
2	9.5	105.3	10	100
1	19	52.6	20	50
0	38	26.3	40	25
16	100	10.0	100	10
15	200	5.0	200	5
14	400	2.5	400	2.5
13	500	2.0	500	2
12	714	1.4	750	1.3
11	1000	1.0	1000	1

3. Data size differs for elapsed value and target value

   FP0: 24 bits

   FP0 compatibility mode: 32 bits

4. F144_TRNS, serial data communication

   When sending data, note the following differences:

Item	FP0	FP0 compatibility mode
Send buffer processing	The send buffer stores the number of bytes to be sent. This number is decremented after every 1-byte transmission.	The number of bytes to be sent remains unchanged during transmission. After transmission has been completed, 0 is written to the send buffer.
Restrictions on the number of bytes to be sent	None	2048 bytes

5. F169_PulseOutput_Jog, JOG operation

   There are two differences between the FP0 and the FP0R specifications:

   Count mode: The FP0R does not support the "no counting" setting. Instead, incremental counting is performed with the FP0 pulse output instructions set to "no

counting".

Pulse width specification: In the FP0 compatibility mode, the duty ratio is fixed at 25%. Differing settings in the FP0 programs will be ignored.

6. F168_PulseOutput_Home, Home return

   In FP0 compatibility mode, the elapsed value is counted during home return operations. With the FP0, the elapsed value is indefinite. In both cases, the elapsed value will be reset to 0 when home return has been completed.

7. Real number calculation process

   Since the accuracy of real number calculation has been improved, the calculation results obtained in the FP0 compatibility mode may differ from the results obtained in the existing FP0 program.

8. If the secondary battery installed in the T32 type is out of charge, the next power-on process will be different:

   FP0: The value in the hold area of the data memory will be unstable.

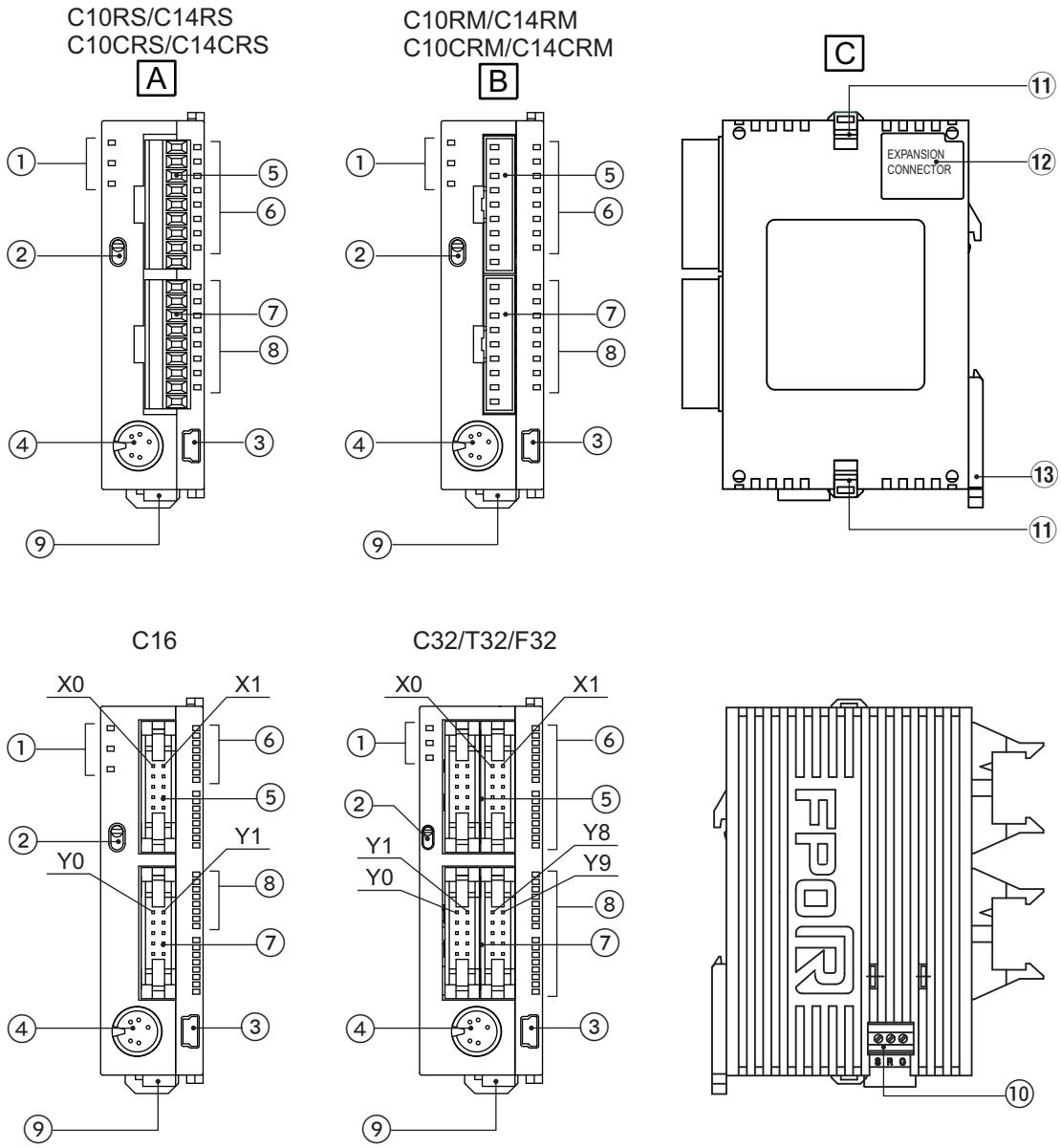
   FP0 compatibility mode: The value in the hold area of the data memory will be cleared to 0.

9. The sampling trace function is not available in FP0 compatibility mode.

Chapter 3

CPU Types

3.1 Parts and Functions, CPU



A	Terminal block type
B	MIL connector type
C	Right side view (common to all CPU types)

① Operation status LEDs

Display the current operation mode or the occurrence of an error.

LED	Description
RUN (green)	Lights when in RUN mode and indicates that the program is being executed.
	Flashes during forced input/output (RUN and PROG. LEDs flash alternately).
PROG. (green)	Lights when in PROG mode and indicates that operation has stopped.
	Flashes during forced input/output (RUN and PROG. LEDs flash alternately).
ERROR/ALARM (red)	Flashes when an error is detected by the self-diagnostic function (ERROR).
	Lights if a hardware error occurs, or if operation slows because of the program, and the watchdog timer is activated (ALARM).

② Operation mode selector

Used to change the operation mode of the PLC.

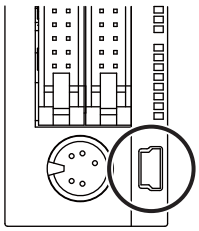
Switch position	Operation mode
RUN (upward)	Sets RUN mode. The program is executed and operation begins.
PROG. (downward)	Sets PROG mode. Operation stops. In this mode, programming via the TOOL port is possible.

When performing remote switching with the programming tool, the position of the operation mode selector and the actual operation mode may differ. Verify the mode with the operation status LED. Otherwise, restart the FP0R and set the operation mode using the operation mode selector.

③ USB port (5-pin Mini-B type)

Used to connect a programming tool.

Panasonic's USB cable CABMINIUSB5D or a commercial USB2.0 AB type cable can be used.

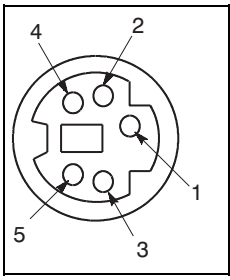


To use the USB port, you must install the USB driver (see page 87).

④ TOOL port (RS232C)

Used to connect a programming tool.

A commercial 5-pin mini DIN connector is used for the TOOL port on the CPU.



Pin no.	Signal name	Abbreviation	Signal direction
1	Signal Ground	SG	–
2	Send Data	SD	CPU → External device
3	Receive Data	RD	CPU ← External device
4	(Not used)	–	–
5	+5V	+5V	CPU → External device

The factory settings are shown below. They can be changed in the system registers.

Communication parameter	Factory settings
Baud rate	9600bit/s
Data length	8
Parity	Odd
Stop bit	1bit

Set the station number for the TOOL port in the TOOL port setting area of the system registers.

⑤ Input connector

⑥ Input status LEDs

⑦ Output connector

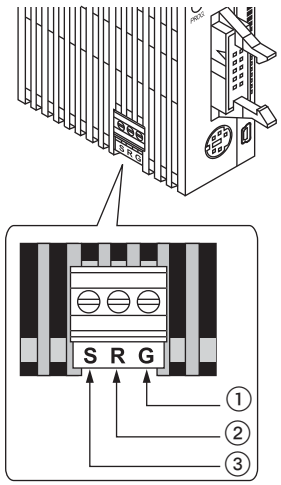
⑧ Output status LEDs

⑨ Power supply connector (24V DC)

Use the power supply cable provided. Product no.: AFPG805

⑩ COM port (RS232C) (C10CR, C14CR, C16C, C32C, T32C, F32C)

This port is used to connect devices via RS232C to enable data input/output.



①	Signal Ground
②	Receive Data (Input)
③	Send Data (Output)

⑪ Expansion hook

Used to secure an expansion unit. The hook is also used for installation on the flat type mounting plate (part no. AFP0804).

⑫ Connector for FP0 expansion units

Connects an FP0 expansion unit to the internal circuit. The connector is located under the seal.

⑬ DIN rail attachment lever

Used for easy attachment to a DIN rail. The lever is also used for installation on a slim type mounting plate. See "Using Optional Mounting Plates" on page 59.

3.2 Input Specifications, CPU

The input specifications below apply to all FP0R CPU types.

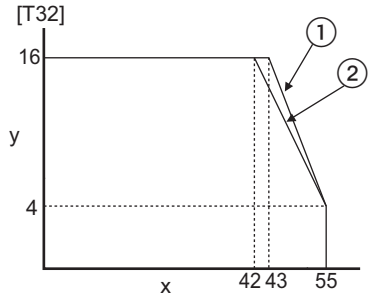
Item		Description
Insulation method		Optical coupler
Rated input voltage		24V DC
Operating voltage range		21.6–26.4V DC
Rated input current		≈2.6mA
Input points per common		C10: 6 C14, C16: 8 C32, T32, F32: 16 (Either the positive or negative pole of the input power supply can be connected to the common terminal.)
Min. ON voltage/min. ON current		19.2V DC/2mA
Max. OFF voltage/max. OFF current		2.4V DC/1.2mA
Input impedance		9.1kΩ
Response time	**FALSE → TRUE**	≤20μs (see note)
	TRUE → FALSE	An input time constant (0.1ms–64ms) can be set using the system registers.
Operation mode indicator		LEDs

◆NOTE

This specification applies when the rated input voltage is 24V DC and the temperature is 25°C.

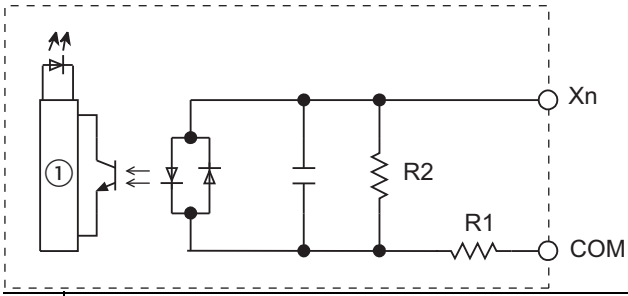
Limitations on the number of input points that are TRUE simultaneously

Keep the number of input points per common which are simultaneously TRUE within the following range as determined by the ambient temperature.



x	Ambient temperature [°C]
y	Number of points per common which are simultaneously TRUE
①	At 24V DC
②	At 26.4V DC

Internal circuit diagram



①	Internal circuit
R1	9.1kΩ
R2	1kΩ

3.3 Output Specifications, CPU

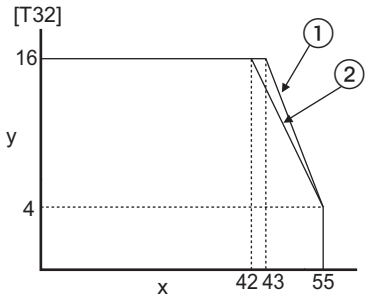
Transistor types

These output specifications apply to the CPU types C32 and C28.

Item		Description	
		NPN	PNP
Insulation method		Optical coupler	
Output type		Open collector	
Rated load voltage		5V DC–24V DC	24V DC
Operating load voltage range		4.75–26.4V DC	21.6–26.4V DC
Max. load current		0.2A	
Output points per common		C16: 8 C32, T32, F32: 16	
OFF state leakage current		≤1μA	
ON state voltage drop		≤0.2V DC	
Response time	FALSE → TRUE	≤20μs (Load current: ≥5mA) ≤0.1ms (Load current: ≥0.5mA)	
	TRUE → FALSE	≤40μs (Load current: ≥5mA) ≤0.2ms (Load current: ≥0.5mA)	
External power supply for driving internal circuit (+ and - terminals)	Voltage	21.6–26.4V DC	
	Current	C16: ≤30mA C32, T32, F32: ≤60mA	C16:≤35mA C32, T32, F32: ≤70mA
Surge absorber		Zener diode	
Operation mode indicator		LEDs	

Limitations on the number of output points which are simultaneously TRUE

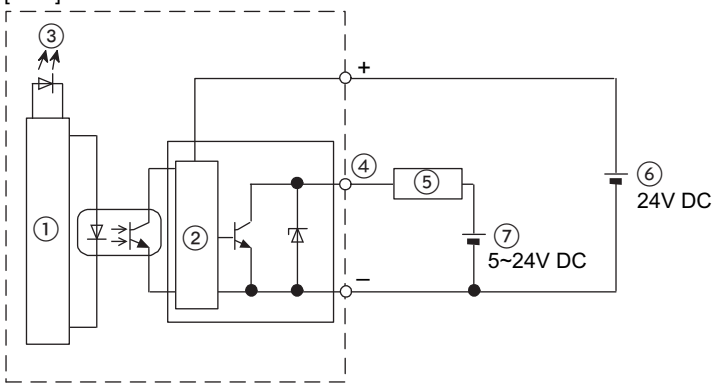
Keep the number of output points per common which are simultaneously TRUE within the following range as determined by the ambient temperature.



x	Ambient temperature [°C]
y	Number of points per common which are simultaneously TRUE
①	At 24V DC
②	At 26.4V DC

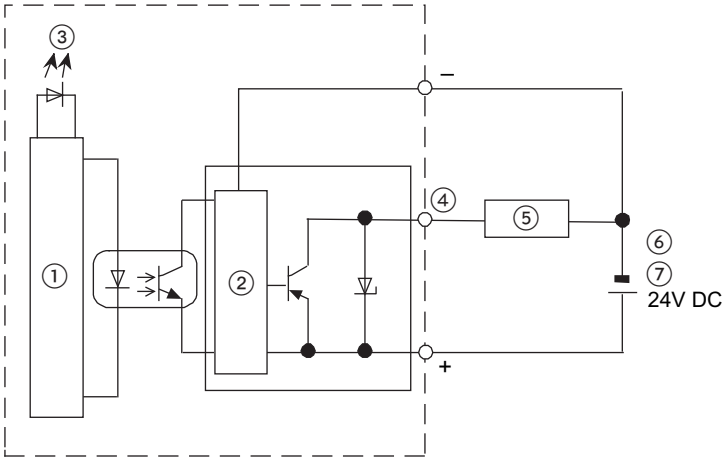
Internal circuit diagram

[NPN]



①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output indicator LED	⑦	Load power supply
④	Output		

[PNP]

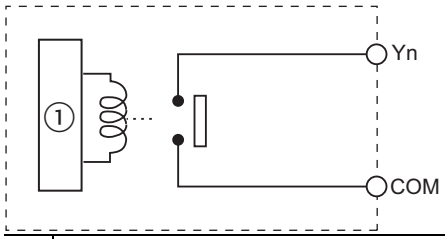


①	Internal circuit	⑤	Load
②	Output circuit	⑥	External power supply
③	Output indicator LED	⑦	Load power supply
④	Output		

Relay types (C10/C14)

Item		Description
Output type		1a output
Nominal switching capacity (resistive load)		2A 250V AC, 2A 30V DC (≤4.5A/common)
Output points per common		C10: 2+1+1 C14: 4+1+1
Response time	**FALSE → TRUE**	≈10ms
	TRUE → FALSE	≈8ms
Mechanical lifetime		≥20 000 000 operations (switching frequency: 180 operations/min)
Electrical lifetime		≥100 000 operations (switching frequency at nominal switching capacity: 20 operations/min)
Surge absorber		–
Operation mode indicator		LEDs

Internal circuit diagram

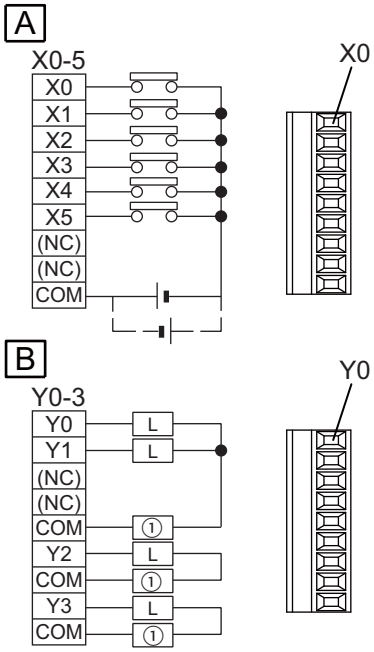


①	Internal circuit

3.4 Terminal Layout

3.4.1 C10 CPU

C10RS, C10CRS, C10RM, C10CRM

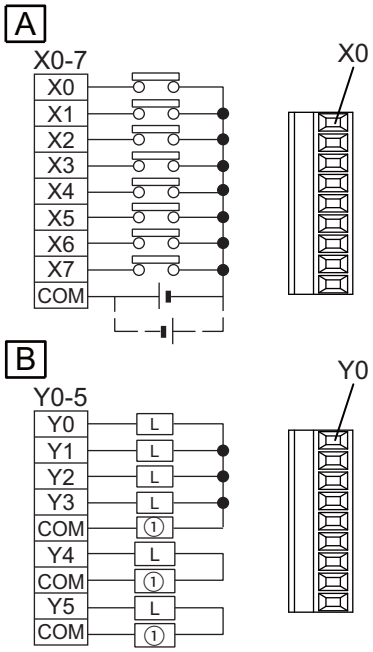


(The above illustration is the terminal block type.)

A	Input
B	Output
①	Power supply

3.4.2 C14 CPU

C14RS, C14CRS, C14RM, C14CRM



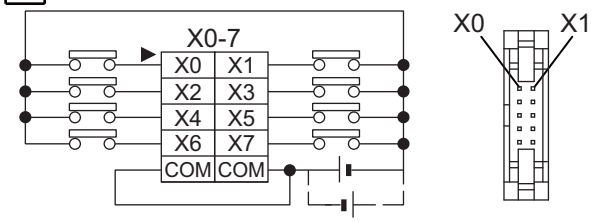
(The above illustration is the terminal block type.)

A	Input
B	Output
①	Power supply

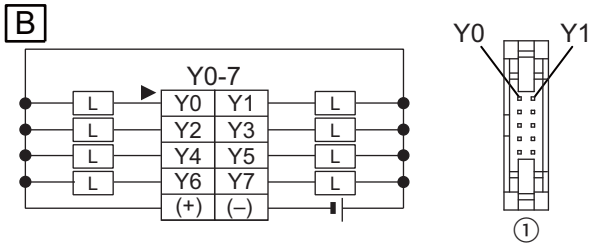
3.4.3 C16 CPU

C16T, C16CT

A

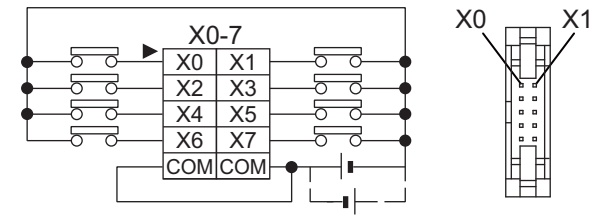


B

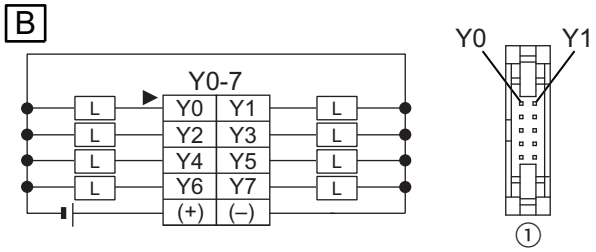


C16P, C16CP

A



B

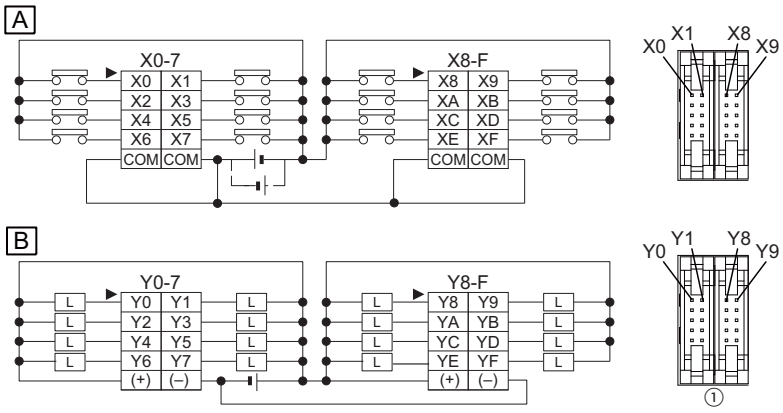


The COM terminals of the input circuits are connected internally.

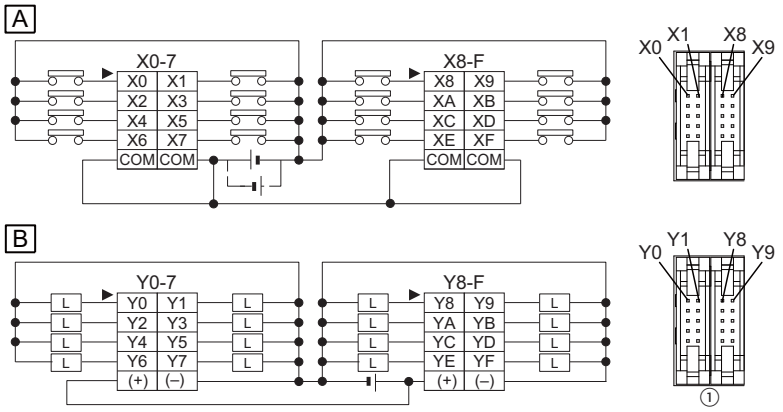
A	Input
B	Output
①	Connector front view

3.4.4 C32 CPU

C32T, C32CT, T32CT, F32CT



C32P, C32CP, T32CP, F32CP



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

A	Input
B	Output
①	Connector front view

3.5 Backup and Clock/Calendar Functions

The FP0R-T32 CPU is equipped with a secondary battery (charging type). This battery makes it possible to use:

- additional hold areas for data registers or other data
- the clock/calendar function

The FP0R-F32 CPU has a built-in FRAM, which allows saving all data without a backup battery. The FP0R-F32 type does not offer a clock/calendar function.

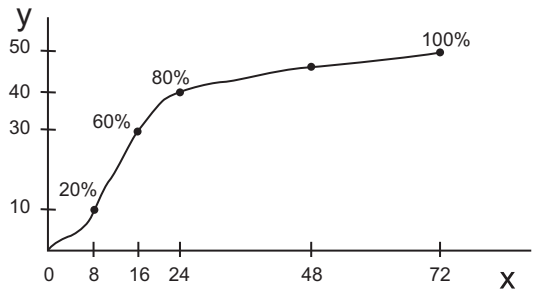
Charging the battery

The built-in backup battery is not charged when the unit is shipped. Charge the battery sufficiently before use.

Charging time for full charge: 72 hours (at an ambient temperature of 25°C)

The battery will be charged automatically when the DC power is supplied to the CPU.

The number of days the backup battery remains effective (backup time) depends on the charging time. If the battery has been fully charged (72 hours at an ambient temperature of 25°C), the battery will function for approx. 50 days.



x	Charging time (hours)
y	Backup time (days at 25°C)

The backup time will vary according to the ambient temperature when the battery is charged.

Ambient temperature when charged	Backup time
70°C	≈14 days
25°C	≈50 days
-20°C	≈25 days

Predicted life of built-in backup battery

The life of the built-in backup battery varies depending on the ambient temperature while the CPU is on (energized).

◆**NOTE**

The temperature when the CPU is off (not powered) has little influence on the battery life.

Ambient temperature	Lifetime of built-in backup battery
55°C	≈430 days (≈1 year)
45°C	≈1200 days (≈3 years)
40°C	≈2100 days (≈6 years)
35°C	≈3300 days (≈9 years)
≤34°C	≈10 years

The built-in backup battery cannot be replaced.

Precision of clock/calendar

Ambient temperature	Error
0°C	<104s/month
25°C	<51s/month
55°C	<155s/month

3.5.1 Backup Function

Additional hold areas which will be saved with a backup battery (FP0R-T32) or with the built-in FRAM (FP0R-F32) can be specified for the following memory areas:

- Timers/Counter (T/C)
- Internal relays (R)
- Data registers (DT)
- Step ladders

Programs and system register settings will be held in the internal ROM regardless of the built-in backup battery.

Specifying hold areas

If no settings are made in system registers 6 to 14, the default address ranges will be saved when the PLC is turned off. To save additional hold areas, follow the procedure below.

◆**Procedure**

1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**

3. **Double-click "Hold On/Off"**

◆CAUTION

If the battery is empty and additional hold areas have been defined, the hold/non-hold operation becomes unstable. The data value will become indefinite. It is cleared to 0 the next time the power is turned on.

Do not forget to monitor the battery status or to reset the hold areas to the default values if no battery is used.

An empty battery status is indicated by the following:

- Special internal relays R9005 and R9006 will turn to TRUE if the battery voltage drops. The relays can be evaluated using the system variables sys_bIsBatteryErrorHold and sys_bIsBatteryErrorNonHold.
- The ERROR/ALARM LED will flash if the battery voltage drops.

We recommend adding a program for clearing the data to 0 when the values in the hold areas become indefinite.

3.5.2 Clock/Calendar Function

As the initial clock/calendar values are unstable, write the values using a programming tool.

3.5.2.1 Memory Area for Clock/Calendar Function

With the clock/calendar function, clock and calendar data stored in special data registers DT90053 to DT90057 can be read and used in sequence programs. To access special data registers and special internal relays, use the PLC-independent system variables.

X: Available, –: not available

Special data register	FPWIN Pro system variable	Upper byte	Lower byte	Reading	Writing
DT90053	sys_w_RTC_HourMin	Hour data 16#00–16#23	Minute data 16#00–16#59	X	–
DT90054	sys_w_RTC_MinSec	Minute data 16#00–16#59	Second data 16#00–16#59	X	X
DT90055	sys_w_RTC_DayHour	Day data 16#01–16#31	Hour data 16#00–16#23	X	X
DT90056	sys_w_RTC_YearMonth	Year data 16#00–16#99	Month data 16#01–16#12	X	X
DT90057	sys_w_RTC_DayOfWeek	–	Day-of-the-week data 16#00–16#06	X	X
DT90058	sys_w_RTC_Set	Bit 15=TRUE (16#8000): activates clock/calendar setting Bit 0=TRUE (16#0): sets seconds to 0		X	X

3.5.2.2 Settings for Clock/Calendar Function

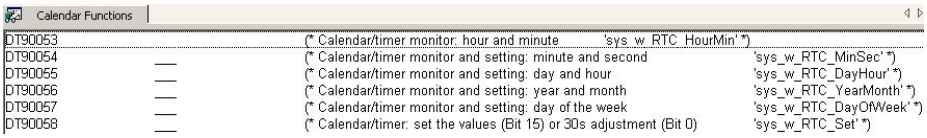
The clock/calendar values are backed up using a battery. Therefore, the clock/calendar function cannot be used unless a battery has been installed.

There are no default clock/calendar settings, so the programming tool or another means must be used to specify these values. There are two ways to set the clock/calendar function:

Using the programming software

1. **Online → Online Mode or**
2. **Monitor → Display Special Registers → Calendar Functions**
3. **Enter the desired date and time values**

   Confirm each value with [Enter].


Calendar Functions
DT90053 (* Calendar/timer monitor: hour and minute 'sys_w_RTC_HourMin' *)
DT90054 (* Calendar/timer monitor and setting: minute and second 'sys_w_RTC_MinSec' *)
DT90055 (* Calendar/timer monitor and setting: day and hour 'sys_w_RTC_DayHour' *)
DT90056 (* Calendar/timer monitor and setting: year and month 'sys_w_RTC_YearMonth' *)
DT90057 (* Calendar/timer monitor and setting: day of the week 'sys_w_RTC_DayOfWeek' *)
DT90058 (* Calendar/timer: set the values (Bit 15) or 30s adjustment (Bit 0) 'sys_w_RTC_Set' *)

Using a program

1. The date/time values are written to special data registers DT90054 to DT90057.
2. A value of 16#8000 is written to DT90058.

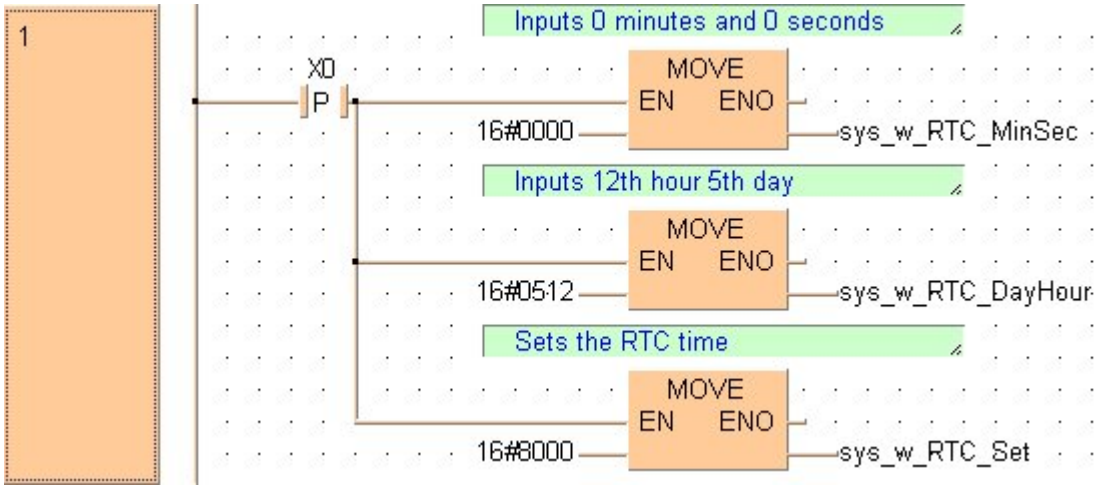
◆NOTE

- **The values can be set using the rising edge signal “P” or by changing 16#8000 to 16#0000.**
- **To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. Please refer to the FPWIN Pro online help for detailed information on using system variables.**
- **To set the clock/calendar, you can also use the instruction SET_RTC_DT. Please refer to the FPWIN Pro online help for details and a programming example.**

◆EXAMPLE

Set the time to 12:00:00 on the 5th day when X0 turns to TRUE. In this example, the values of DT90054, DT90055, and DT90058 are written using system variables.

LD Body



3.5.2.3 Sample Program for Fixed Schedule and Automatic Start

In this example, the clock/calendar function is used to output the Y0 signal for one second at 8:30 a.m. every day. Here, the hour/minute data stored in special data register DT90053 is used to output the signal at the appointed time. The value of DT90053 is written using a system variable.

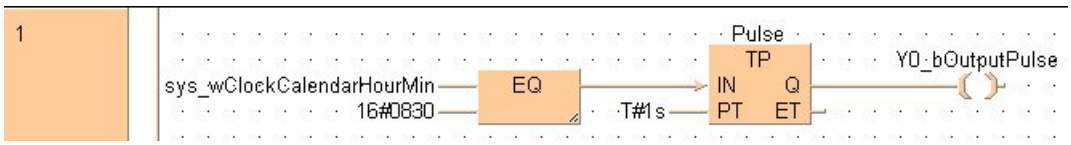
GVL

	Class	Identifier	FP Address	IEC Address	Type	Initial
0	VAR_GLOBAL	Y0_bOutputPulse	Y0	%QX0.0	BOOL	FALSE

POU Header

	Class	Identifier	Type	Initial
0	VAR	Pulse	TP	
1	VAR_EXTERNAL	Y0_bOutputPulse	BOOL	FALSE
2	VAR			

LD Body



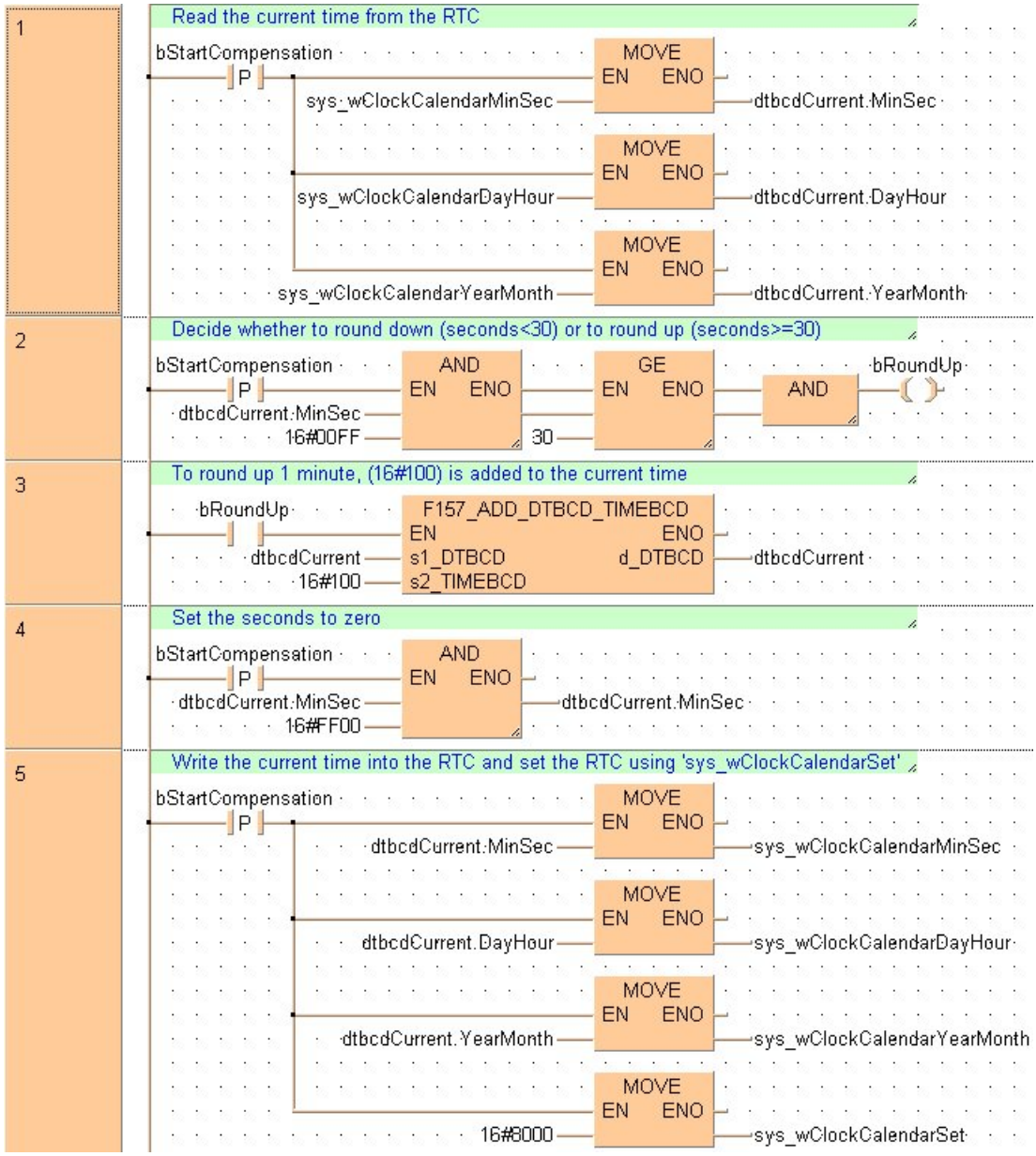
3.5.2.4 Sample Program for 30-Second Compensation

This is a program to perform the compensation for 30 seconds when R0 turns to TRUE. If the 30-second compensation is required, use this program.

POU Header

	Class	Identifier	Type	Initial
0	VAR	bStartCompensation	BOOL	FALSE
1	VAR	bRoundUp	BOOL	FALSE
2	VAR	dtbcdCurrent	DTBCD	
3	VAR	wSec	WORD	0

LD Body



ST Body

```
       if (DF(bStartCompensation)) then
            (* Read the current time from the RTC *)
            dtbcdCurrent.MinSec:=sys_wClockCalendarMinSec;
            dtbcdCurrent.DayHour:=sys_wClockCalendarDayHour;
            dtbcdCurrent.YearMonth:=sys_wClockCalendarYearMonth;

            (* Decide whether to round up (seconds>=30) *)
            if ((dtbcdCurrent.MinSec AND 16#00FF)>30) then
                  (* To round up 1 minute (16#100) is added to the
current time *)
                  F157_ADD_DTBCD_TIMEBCD(s1_DTBCD := dtbcdCurrent,
                                                    s2_TIMEBCD :=
16#100,
                                                    d_DTBCD =>
dtbcdCurrent);
            end_if;
            (* Set the seconds to zero *)
            dtbcdCurrent.MinSec:=dtbcdCurrent.MinSec AND 16#FF00;
            (* Write the current time into the RTC and set the RTC
using 'sys_wClockCalendarSet' *)
            sys_wClockCalendarMinSec:=dtbcdCurrent.MinSec;
            sys_wClockCalendarDayHour:=dtbcdCurrent.DayHour;
            sys_wClockCalendarYearMonth:=dtbcdCurrent.YearMonth;
            sys_wClockCalendarSet:=16#8000;
end_if;
```

Chapter 4

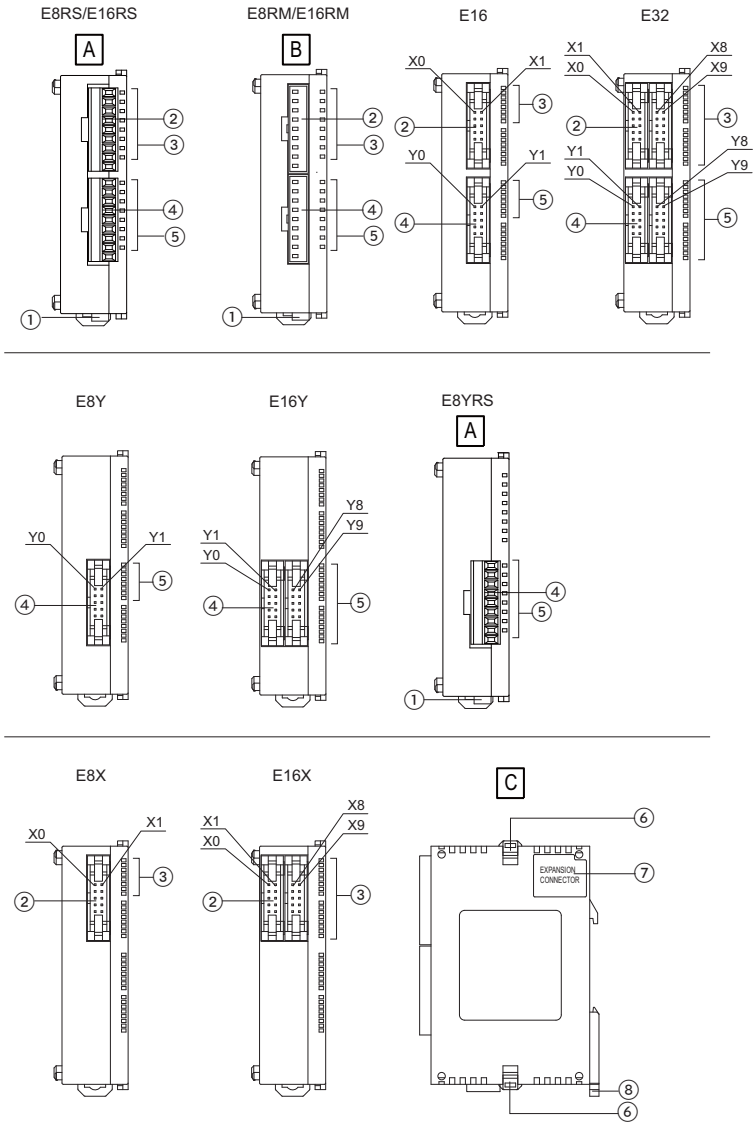
Expansion

4.1 Expansion Method

You can expand the FP0R by adding FP0 I/O expansion units (see page 9), FP0 intelligent units (see page 9) and FP Series link units (see page 10).

The expansion units are connected to the right side of the CPU. Use the expansion connector and the expansion hooks on the side of the unit. See "Connecting FP0 Expansion Units" on page 63.

4.2 Parts and Functions, Expansion Units



A	Terminal block type
B	MIL connector type
C	Right side view (common to all expansion units)

① Power supply connector (24V DC)

Use the power supply cable provided. Product no.: AFP0581

② Input connector

③ Input status LEDs

④ Output connector

⑤ Output status LEDs

⑥ Expansion hook

Used to secure an expansion unit.

⑦ Connector for FP0 expansion units

Connects an FP0 expansion unit to the internal circuit. The connector is located under the seal.

⑧ DIN rail attachment lever

Used for easy attachment to a DIN rail. The lever is also used for installation on a slim type mounting plate. See "Slim Type Mounting Plate" on page 59.

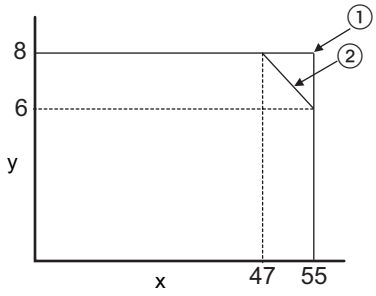
4.3 Input Specifications, Expansion Units

Item		Description
Insulation method		Optical coupler
Rated input voltage		24V DC
Rated input current		≈4.3mA (at 24V DC)
Input impedance		≈5.6kΩ
Operating voltage range		21.6–26.4V DC
Input points per common		E8X/E16P/E16T/E32RS: 8 E32T/E16X: 16 E8R: 4 (Either the positive or negative pole of the input power supply can be connected to the common terminal.)
Min. ON voltage/min. ON current		19.2V DC/3mA
Max. OFF voltage/max. OFF current		2.4V DC/1mA
Response time	**FALSE → TRUE**	≤2ms
	TRUE → FALSE	
Operation mode indicator		LEDs

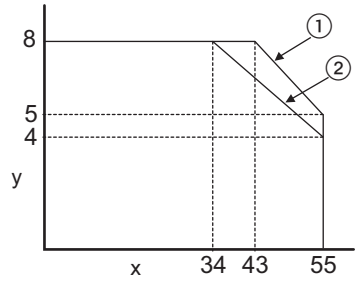
Limitations on the number of input points that are TRUE simultaneously

Keep the number of input points per common which are simultaneously TRUE within the following range as determined by the ambient temperature.

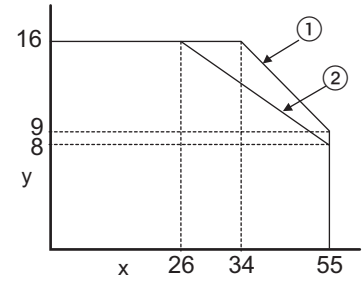
E16R/E32RS



E8X/E16T



E16X/E32T



x	Ambient temperature [°C]
y	Number of points per common which are simultaneously TRUE
①	At 24V DC
②	At 26.4V DC

4.4 Output Specifications, Expansion Units

Relay output specifications (E8RS/E8RM/E8YRS/E16RS/E16RM/E32RS)

Item		Description
Output type		1a
Nominal switching capacity (resistive load)		2A 250V AC, 2A 30V DC (≤4.5A/common)
Output points per common		E8R: 4 E16R/E8YR/E32RS: 8
Response time	**FALSE → TRUE**	≈10ms
	TRUE → FALSE	≈8ms
Mechanical lifetime		≥20 000 000 operations (switching frequency: 180 operations/min)
Electrical lifetime		≥100 000 operations (switching frequency at nominal switching capacity: 20 operations/min)
Surge absorber		–
Operation mode indicator		LEDs

Transistor output specifications

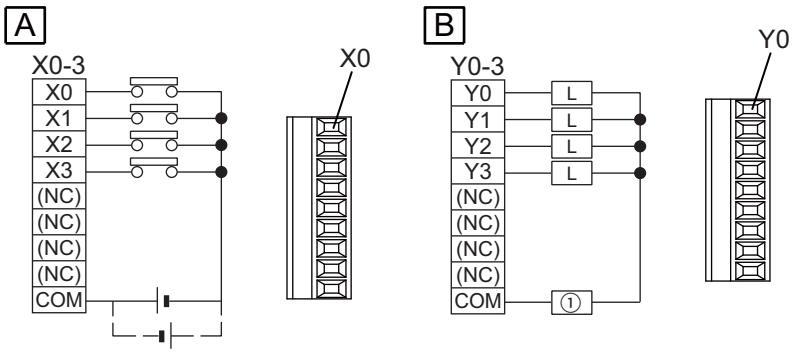
(NPN output type: E8YT/E16YT/E16T/E32T)

(PNP output type: E8YP/E16YP/E16P/E32P)

Item		Description	
		NPN	PNP
Insulation method		Optical coupler	
Output type		Open collector	
Rated load voltage		5V DC–24V DC	24V DC
Operating load voltage range		4.75–26.4V DC	21.6–26.4V DC
Max. load current		0.1A/point (max. 1A/common)	
Max. surge current		0.3A	
Output points per common		E16T/E8Y: 8 E32/E16Y: 16	
OFF state leakage current		≤100μA	
ON state voltage drop		≤1.5V	
Response time	FALSE → TRUE	≤1ms	
	TRUE → FALSE	≤1ms	
External power supply for driving internal circuit	Voltage	21.6–26.4V DC	
	Current	3mA/point	
Surge absorber		Zener diode	
Operation mode indicator		LEDs	

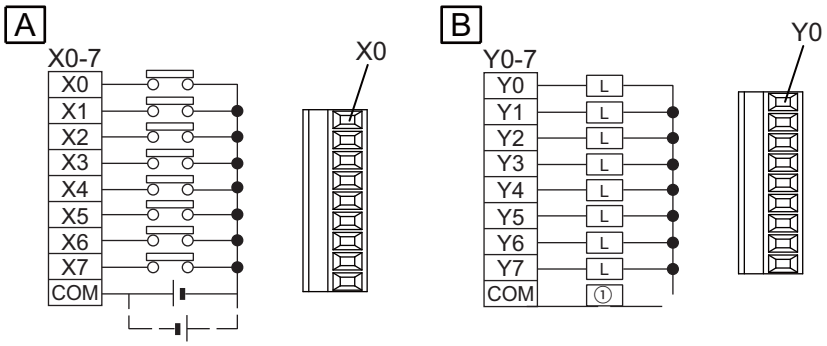
4.5 Terminal Layout

E8RS, E8RM



A	Input
B	Output
①	Power supply

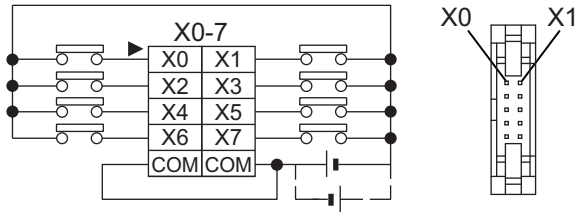
E16R, E8YRS, E32RS



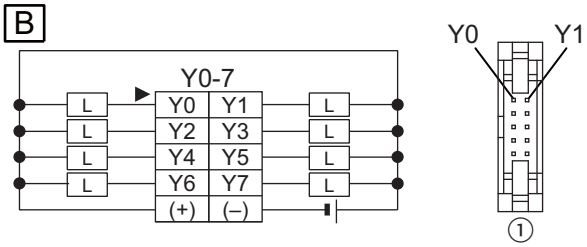
A	Input (no input for E8YRS)
B	Output
①	Power supply

E8X, E16T, E8YT

A



B

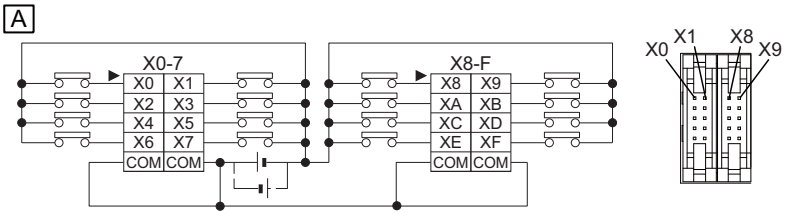


The COM terminals of the input circuits are connected internally.

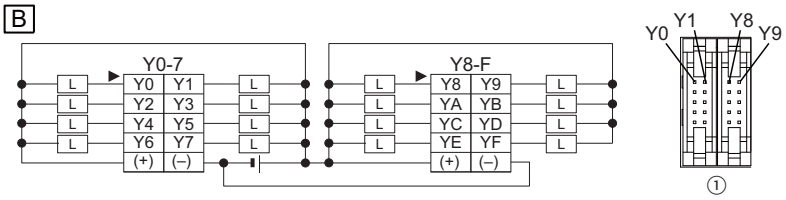
A	Input (no input for E8YT)
B	Output (no output for E8X)
①	Connector front view

E16X, E32T, E16YT

A



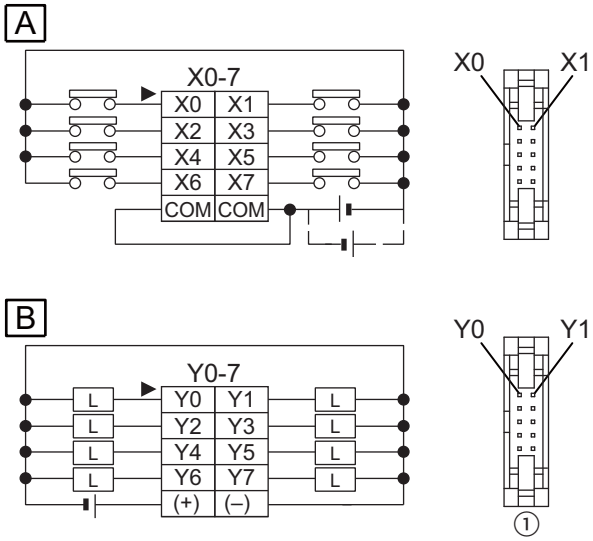
B



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

A	Input (no input for E16YT)
B	Output (no output for E16X)
①	Connector front view

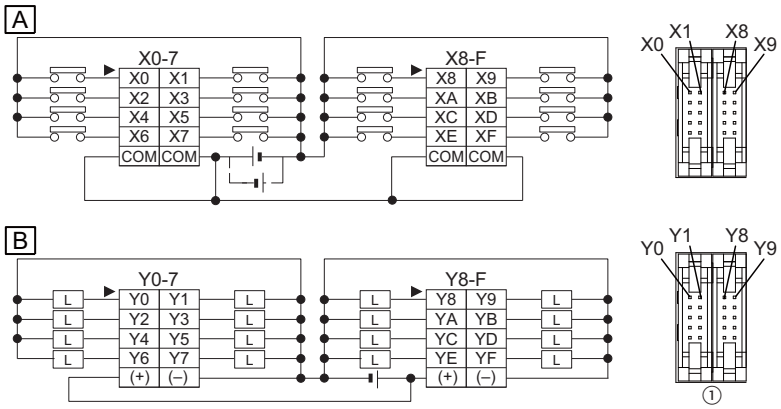
E16P, E8YP



The COM terminals of the input circuits are connected internally.

A	Input (no input for E8YT)
B	Output
①	Power supply

E32P, E16YP



The (+) terminals as well as the (-) terminals of the output circuits are connected internally.

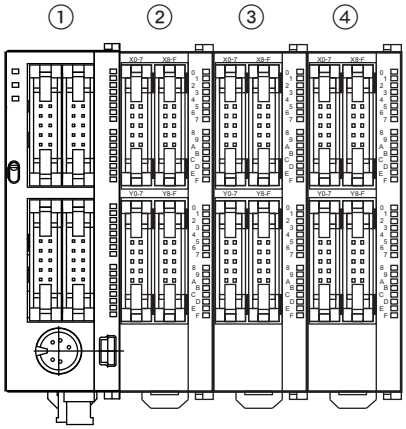
A	Input (no input for E16YP)
B	Output
①	Connector front view

Chapter 5

I/O Allocation

5.1 General

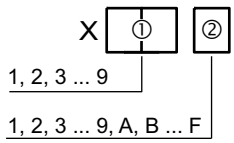
I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location. The I/O allocation of the FP0R CPU is fixed.



Type of unit	Unit number		I/O addresses
FP0R CPU	①	–	X0–XF Y0–YF
FP0 I/O expansion unit	②	1	X20–X3F Y20–Y3F
	③	2	X40–X5F Y40–Y5F
	④	3	X60–X7F Y60–Y7F

◆**NOTE**

- **The input relay "X" and output relay "Y" are expressed as a combination of decimal (①) and hexadecimal (②) numbers:**



- **On the FP0R and the FP0, the same numbers are used for inputs and outputs, e.g. X20, Y20.**
- **The usable I/O numbers depend on the unit type. See "FP0 Expansion Units" on page 52.**

5.2 CPU

The I/O allocation of the FP0R CPU is fixed.

CPU type		Number of I/O points	I/O addresses
C10	**Input**	6	X0–X5
	Output	4	Y0–Y3
C14	**Input**	8	X0–X7
	Output	6	Y0–Y5
C16	**Input**	8	X0–X7
	Output	8	Y0–Y7
C32/T32/F32	**Input**	16	X0–XF
	Output	16	Y0–YF

5.3 FP0 Expansion Units

I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location. Expansion units from the FP0 series are connected on the right side of the CPU. The I/O numbers are allocated from the unit nearest to the CPU in ascending order.

Type of unit			Number of I/O points	Channel	Unit number (installation location)		
					1	2	3
FP0 I/O expansion unit							
	FP0-E8X	Input	8	–	X20–X27	X40–X47	X60–X67
	FP0-E8R	Input	4	–	X20–X23	X40–X43	X60–X63
		Output	4	–	Y20–Y23	Y40–Y43	Y60–Y63
	FP0-E8YR, E8YT, E8YP	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
	FP0-E16X	Input	16	–	X20–X2F	X40–X4F	X60–X6F
	FP0-E16R, E16T, E16P	Input	8	–	X20–X27	X40–X47	X60–X67
		Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
	FP0-E16YT, E16YP	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
	FP0-E32T, E32P, E32RS	Input	16	–	X20–X2F	X40–X4F	X60–X6F
		Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0 analog I/O unit FP0-A21		Input	16	0	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Input	16	1	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
		Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 A/D conversion unit FP0-A80 and FP0 thermocouple unit FP0-TC4, FP0-TC8		Input	16	0, 2, 4, 6	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Input	16	1, 3, 5, 7	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
FP0 D/A conversion unit FP0-A04V, FP0-A04I		Input	16	–	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Output	16	0, 2	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
		Output	16	1, 3	WY3 (Y30–Y3F)	WY5 (Y50–Y5F)	WY7 (Y70–Y7F)
FP0 RTD unit FP0-RTD6		Input	16	0, 2, 4	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Input	16	1, 3, 5	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
		Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 I/O link unit FP0-IOL		Input	32	–	X20–X3F	X40–X5F	X60–X7F
		Output	32	–	Y20–Y3F	Y40–Y5F	Y60–Y7F

◆**NOTE**

The data for each channel of the A/D and D/A conversion units FP0-A80, FP0-TC4/TC8, FP0-A04V/I, and FP0-RTD6 is converted and loaded with a user program that includes a switching flag to convert the data in 16-bit words (see corresponding manuals).

Chapter 6

Installation and Wiring

6.1 Installation

Please follow the installation instructions carefully to prevent failure or malfunctions.

6.1.1 Installation Environment and Space

Operating environment

After installing the unit, make sure to use it within the range of the general specifications:

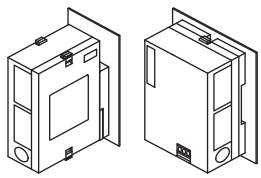
- Ambient temperature: 0–+55°C
- Ambient humidity: 10%–95% RH (at 25°C non-condensing)
- Pollution level: 2
- Do not use the unit in the following environments:
  - Direct sunlight
  - Sudden temperature changes causing condensation
  - Inflammable or corrosive gases
  - Excessive airborne dust, metal particles or salts
  - Benzine, paint thinner, alcohol or other organic solvents or strong alkaline solutions such as ammonia or caustic soda
  - Direct vibration, shock or direct drop of water
  - Influence from power transmission lines, high voltage equipment, power cables, power equipment, radio transmitters, or any other equipment that would generate high switching surges. Maintain at least 100mm of space between these devices and the unit.

Static electricity

- Before touching the unit or equipment, always touch some grounded metal to discharge any static electricity you may have generated (especially in dry locations). The discharge of static electricity can damage parts and equipment.

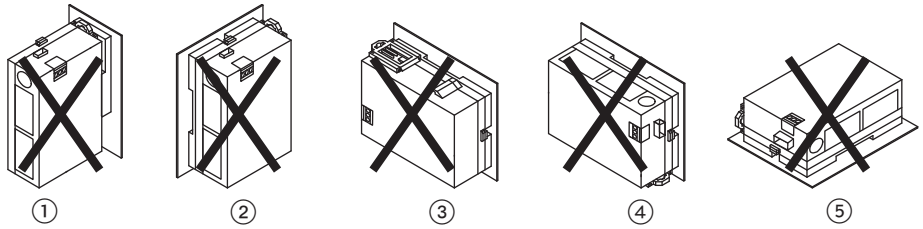
Measures regarding heat discharge

- Always install the CPU orientated with the TOOL port facing outward on the bottom in order to prevent the generation of heat.



OK

- Do **NOT** install the CPU as shown below.

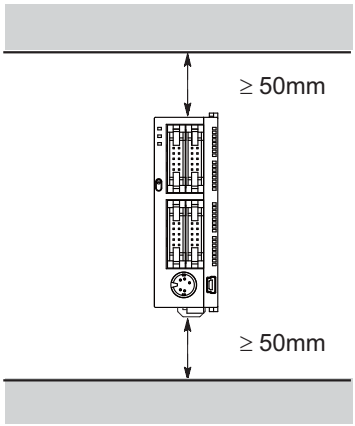


①	Upside-down
②	Upside-down
③	Input and output connectors face down
④	Input and output connectors on top
⑤	Horizontal installation of the unit

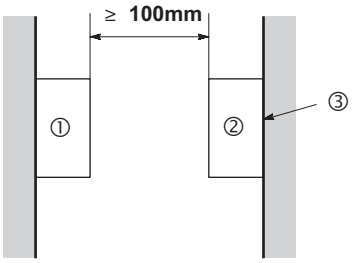
- Do not install the unit above devices which generate heat such as heaters, transformers or large scale resistors.

Installation space

- Leave at least 50mm of space between the wiring ducts of the unit and other devices to allow heat radiation and unit replacement.



- Maintain a minimum of 100mm between devices to avoid adverse affects from noise and heat when installing a device or panel door to the front of the unit.



①	PLC
②	Other device
③	Panel door

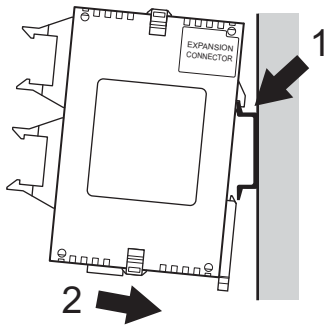
- Keep the first 100 mm from the unit's front surface clear of objects to allow wiring and

the connection of the programming tool.

6.1.2 Using DIN Rails

The CPU can be easily attached to DIN rails.

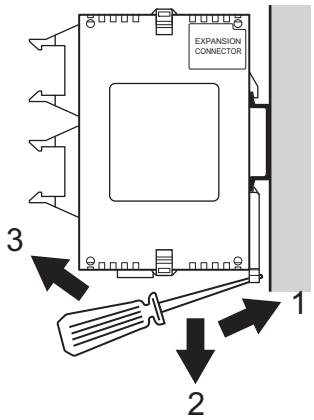
◆Procedure



1. **Fit upper hook of unit onto DIN rail**
2. **Without moving upper hook, press on lower hook to fit unit into position**

Removal is very simple, too:

◆Procedure



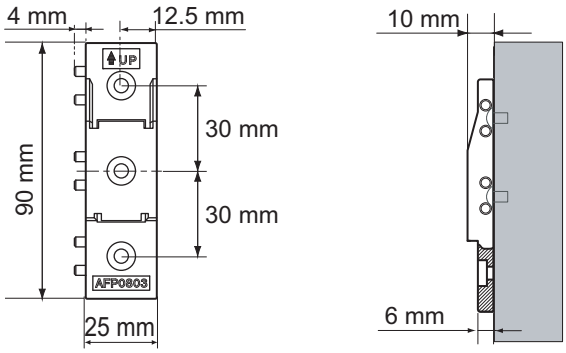
1. **Insert slotted screwdriver into DIN rail attachment lever**
2. **Pull attachment lever downwards**
3. **Lift up unit and remove from rail**

6.1.3 Using Optional Mounting Plates

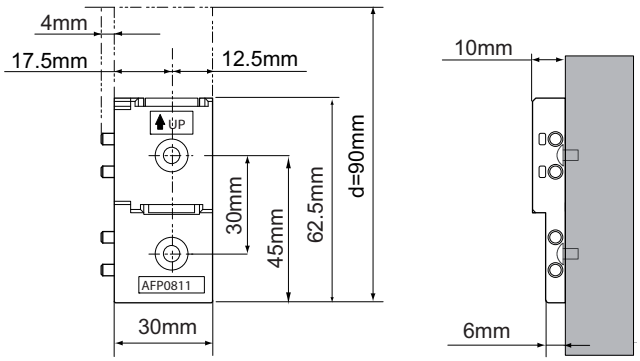
Use M4 size pan-head screws for attaching the mounting plate to the mounting panel. The diagrams below show the dimensions of the mounting plates.

6.1.3.1 Slim Type Mounting Plate

Use AFP0803 for mounting CPU types C10, C14, and C16 as well as 25mm type FP0 units:



Use AFP0811 for mounting CPU types C32, T32, and F32 as well as 30mm type FP0 units:

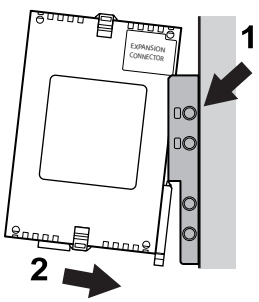


d	Total unit length

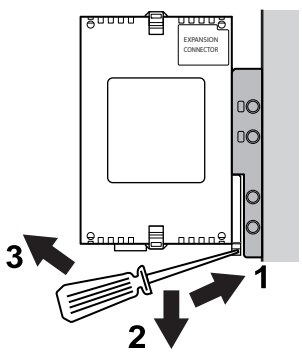
Installation and removal

Installation and removal of the unit is similar to the procedure using DIN rails:

Installation:



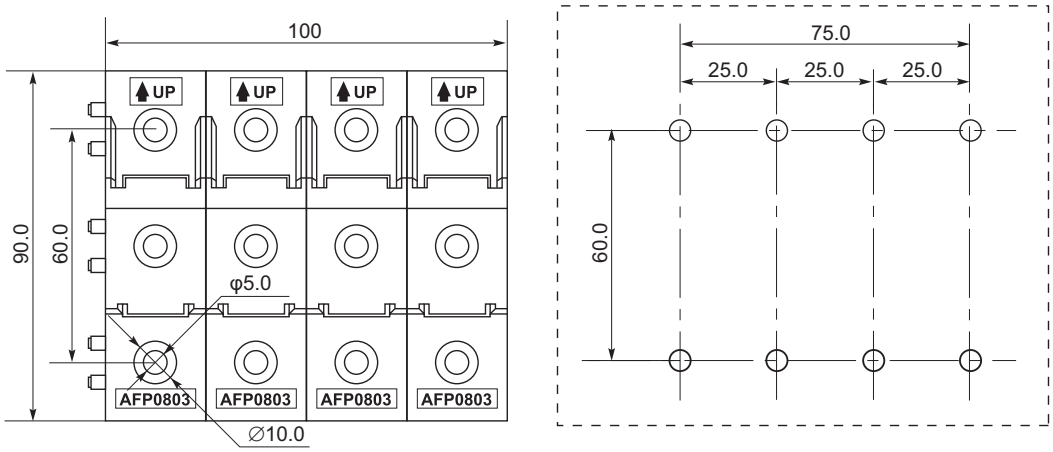
Removal:



Combining mounting plates

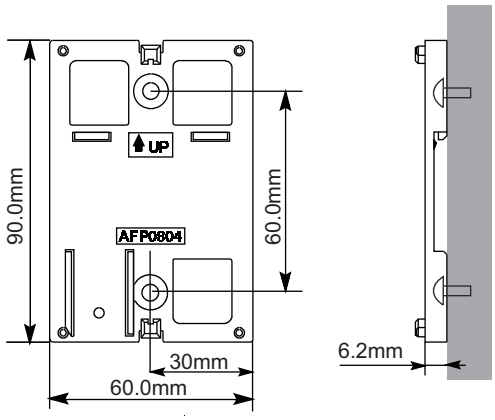
When combining several mounting plates, tighten the screws after joining all of the mounting plates to be connected. Tighten all corner screws.

The following diagram shows the combination of the mounting plates AFP0803 when the maximum number of expansion units is used and the mounting hole dimensions:

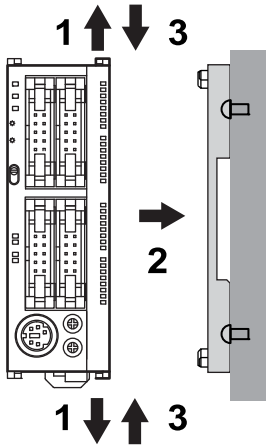


6.1.3.2 Flat Type Mounting Plate

The flat type mounting plate (AFP0804) should only be used with a stand-alone CPU. It should not be used if an expansion unit is attached to the CPU.



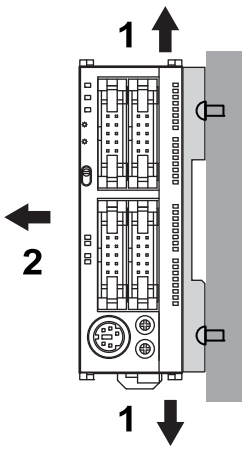
Installation



◆Procedure

1. **Raise expansion hooks on top and bottom of the unit**
2. **Press unit on mounting plate and align expansion hooks with plate**
3. **Push expansion hooks back into place**

Removal

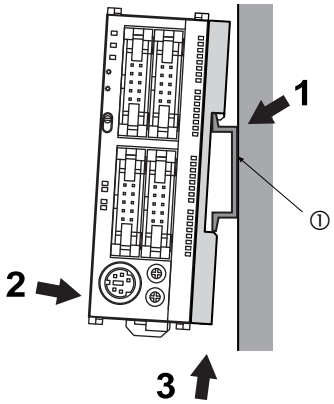


◆Procedure

1. **Raise expansion hooks on top and bottom of the unit**
2. **Remove unit from mounting plate**

Attachment to DIN rail

A unit with an attached flat type mounting plate can also be installed sideways on a DIN rail.



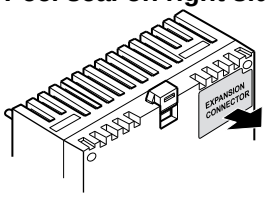
①	DIN rail

6.2 Connecting FP0 Expansion Units

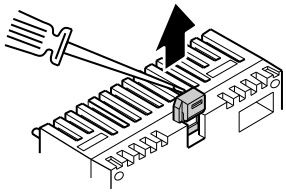
The expansion units are connected to the right side of the CPU. Use the expansion connector and the expansion hooks on the side of the unit.

◆Procedure

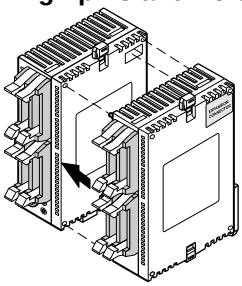
1. **Peel seal on right side of the unit to expose expansion connector**



2. **Raise expansion hooks on top and bottom of the unit**

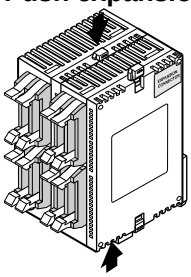


3. **Align pins and holes in all four corners**



4. **Insert pins into holes so that there is no gap between units**

5. **Push expansion hooks back into place**



You can now add up to two more units in the same manner.

6.3 Safety Instructions for Wiring

In certain applications, malfunction may occur for the following reasons:

- Power ON timing differences between the PLC system and input/output or motorized devices
- A response time lag when a momentary power drop occurs
- Abnormality in the PLC, external power supply circuit, or other devices

In order to prevent a malfunction that results in a system shutdown, choose the adequate safety measures listed below:

Interlock circuit

When a motor's clockwise/counter-clockwise operation is controlled, provide an interlock circuit that prevents clockwise and counter-clockwise signals from being input into the motor at the same time.

Emergency stop circuit

Provide an emergency stop circuit externally to turn off the power supply of controlled devices in order to prevent a system shutdown or an irreparable accident if a malfunction occurs.

Start-up sequence

The PLC should be operated only after all of the field devices are energized. To ensure this sequence, the following measures are recommended:

- Turn ON the PLC with the operation mode selector set to PROG mode, and then switch to RUN mode
- Program the PLC so as to disregard the inputs and outputs until the field devices are energized

◆**NOTE**

When stopping the operation of the PLC, also have the input/output devices turned off after the PLC has stopped operating.

Grounding

When installing the PLC next to devices that generate high voltages from switching, such as inverters, do not ground them together. Use an exclusive ground for each device.

Momentary power failures

The FP0R continues to operate normally for a certain period of time in case of a momentary power failure. We call this the momentary power off time. However, if the power failure exceeds this period of time, operation depends on the combination of units, the power supply voltage, etc. In some cases, operation mirrors a power supply reset.

For the momentary power off time values, see "General Specifications" on page 224.

Protection of power supply

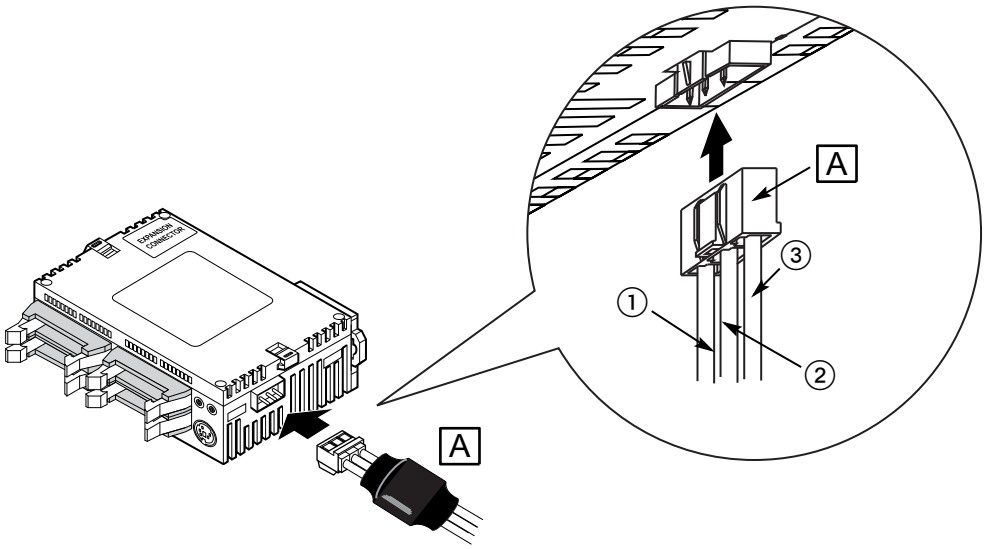
- An insulated power supply with an internal protective circuit should be used (FP0-PSA2 or FP-PS24-050). The power supply for the CPU is a non-insulated circuit, so if an incorrect voltage is directly applied, the internal circuit may be damaged or destroyed.
- If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.

Protection of output sections

If current exceeding the nominal switching capacity is being supplied in the form of a motor lock current or a coil shorting in an electromagnetic device, a protective element such as a fuse should be attached externally.

6.4 Wiring the Power Supply

Use the power supply cable provided. Attach as shown.



A	Power supply cable (AFPG805)
①	Brown: 24V DC
②	Blue: 0V
③	Green: function earth

Specifications

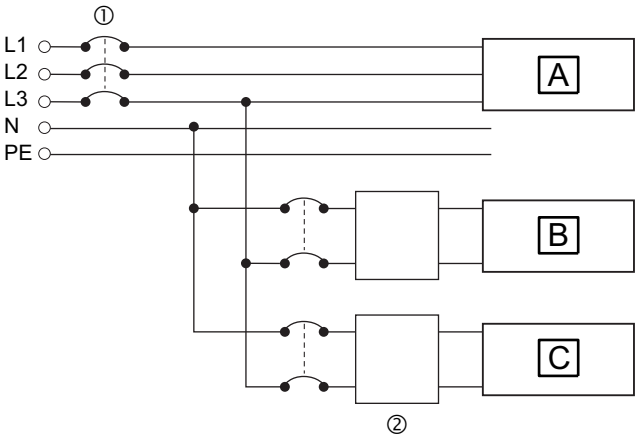
Rated voltage:	24V DC
Operating voltage range:	21.6–26.4V DC

◆**NOTE**

- **To minimize adverse effects from noise, twist the brown and blue wires of the power supply cable.**
- **To protect the system against faulty voltages from the power supply line, use an insulated power supply with an internal protective circuit.**
- **The regulator on the unit is a non-insulated type.**
- **If using a power supply device without an internal protective circuit, always make sure power is supplied to the unit through a protective element such as a fuse.**

Isolation of power supply systems

Isolate the wiring systems to the CPU, input/output devices, and mechanical power apparatus.



A	Mechanical power apparatus
B	Input/output devices
C	CPU
①	Circuit breaker
②	Insulated DC power supply

Power supply sequence

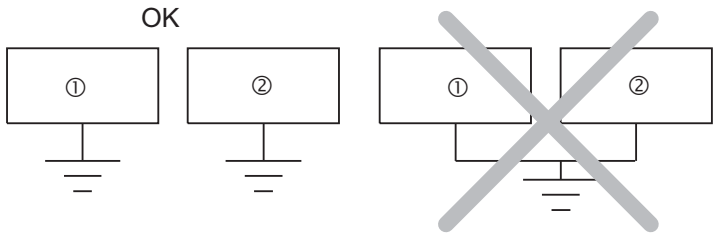
- **Make sure the power supply of the CPU turns off before the power supply for input and output. If the power supply for input and output is turned off first, the CPU will detect the input fluctuations and may begin an unexpected operation.**
- **Be sure to supply power to a CPU and an expansion unit from the same power supply, and turn the power on and off simultaneously for both.**

6.4.1 Grounding

If necessary, ground the instrument to increase the noise resistance.

◆NOTE

- **For grounding purposes, use wiring with a minimum of 2mm^2. The grounding connection should have a resistance of less than 100Ω.**
- **The point of grounding should be as close to the PLC as possible. The ground wire should be as short as possible.**
- **If two devices share a single ground point, it may produce an adverse effect. Always use an exclusive ground for each device.**



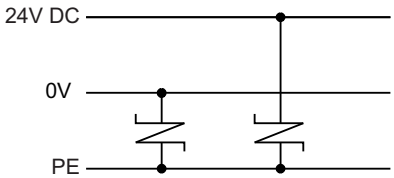
①	PLC
②	Other device (inverter etc.)

Risk of short circuits

Depending on the surroundings in which the equipment is used, grounding may cause problems.

Example 1:

Since the power supply line of the FP0 expansion unit (24V DC and 0V terminal) is connected to the function earth through a varistor, the varistor may be shorted if there is an irregular potential between the power supply line and function earth. (The power supply line of the FP0R is connected to function earth through a high-voltage capacitor. Therefore, there is no risk of a short-circuit.)



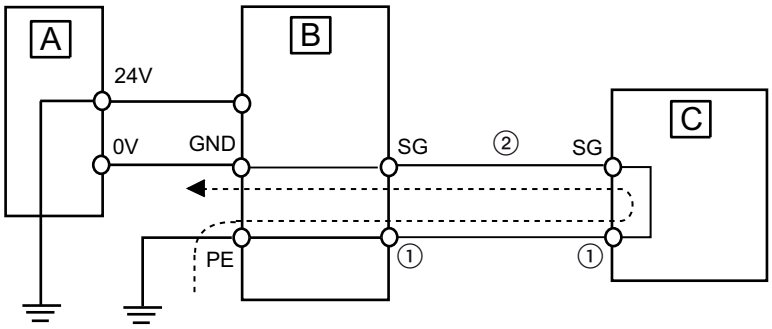
Power supply line of FP0R with built-in 39V varistor

Example 2:

Do not ground the function earth terminal of the FP0R when grounding a plus terminal (+) of the power supply.

In some computers, the SG terminal of the RS232C port is connected with the connector shielding. Also, the FP0R TOOL port shielding is connected with the function earth terminal

(PE). Therefore, the GND and function earth terminals of the FP0R are connected if the computer is connected. Especially when the FP0R is connected to a computer with a plus terminal (+) grounded, the minus terminal (-) of an FP0R is connected with the function earth terminal. A resulting short circuit may damage the FP0R and its neighboring parts.



A	Power supply	①	Shielding
B	CPU	②	Cable
C	Computer		

6.5 Input and Output Wiring

◆**NOTE**

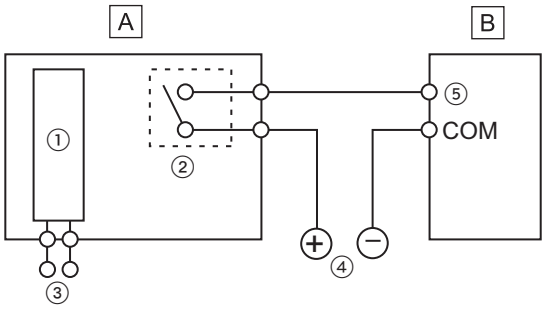
- **Separate the input/output wires from the power and high voltage wires by at least 100mm.**
- **Be sure to select the thickness (diameter) of the input and output wires while taking into consideration the required current capacity.**
- **Arrange the wiring so that the input and output wiring are separated and so that these wirings are separated from the power wiring as much as possible. Do not route them through the same duct or wrap them up together.**

6.5.1 Input Wiring

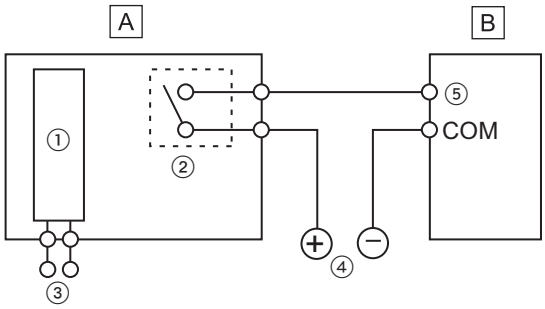
For connecting input devices see the diagrams and recommendations given below.

Relay output type

NPN input



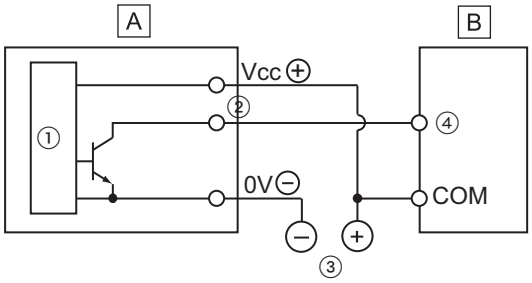
PNP input



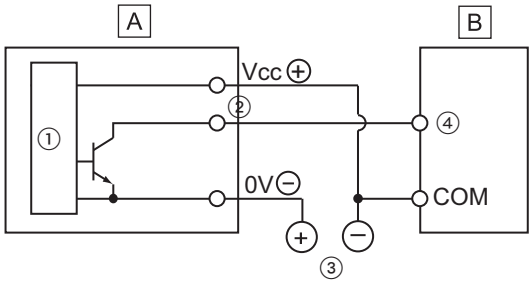
A	Sensor	③	Power supply for sensor
B	FP0R	④	Power supply for input
①	Internal circuit	⑤	Input terminal
②	Relay		

Open collector output type

NPN output

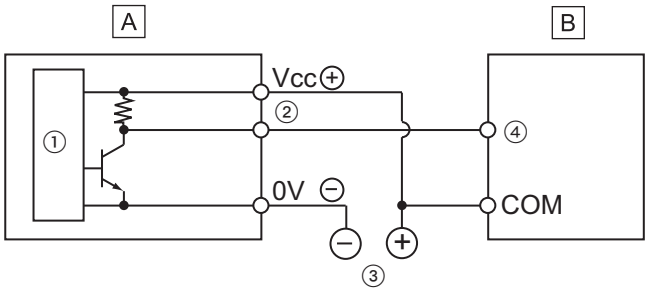


PNP output



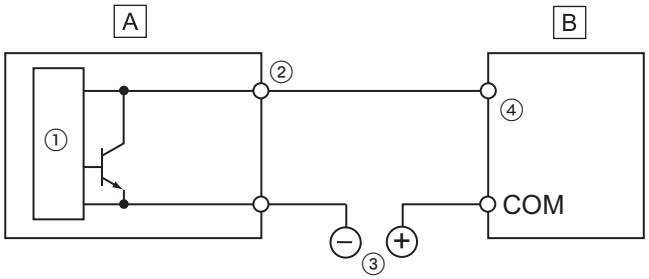
A	Sensor	②	Output
B	FP0R	③	Power supply for input
①	Internal circuit	④	Input terminal

Voltage output (universal output) type



A	Sensor	②	Output
B	FP0R	③	Power supply for input
①	Internal circuit	④	Input terminal

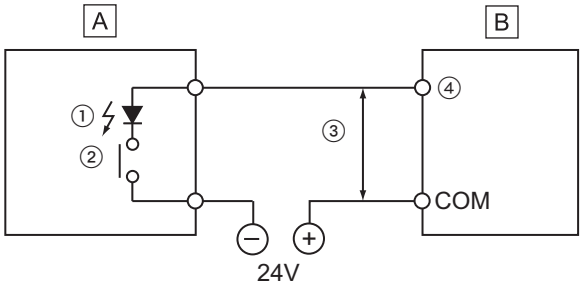
Two-wire output type



A	Sensor	②	Output
B	FP0R	③	Power supply for input
①	Internal circuit	④	Input terminal

Precaution when using an LED-equipped Reed switch

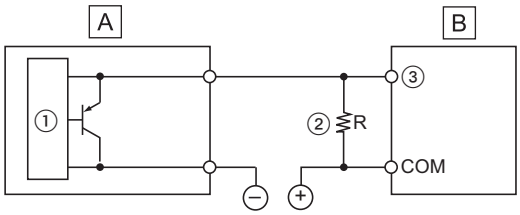
When an LED is connected in series to an input contact such as an LED-equipped Reed switch, make sure that the ON voltage applied to the PLC input terminal is greater than 21.6V DC. In particular, take care when connecting a number of switches in series.



A	LED-equipped Reed switch	②	Contact
B	FP0R	③	≥21.6V
①	LED	④	Input terminal

Precaution when using a two-wire type sensor

If the input of the PLC does not turn off because of leakage current from the two-wire type sensor (photoelectric sensor or proximity sensor), the use of a bleeder resistor is recommended, as shown below.



A	Two-wire type sensor	②	Bleeder resistor
B	FP0R	③	Input terminal
①	Internal circuit		

The formula is based on an input impedance of 9.1kΩ. The input impedance varies depending on the input terminal number.

The off voltage of the input is 2.4V. Therefore, select a bleeder resistor value R so that the voltage between the COM terminal and the input terminal will be less than 2.4V.

$$I \times \frac{9.1R}{9.1R + R} \leq 2.4$$

Therefore:

$$R \leq \frac{21.84R}{9.1I - 2.4} \ (k\Omega)$$

The wattage W of the resistor is:

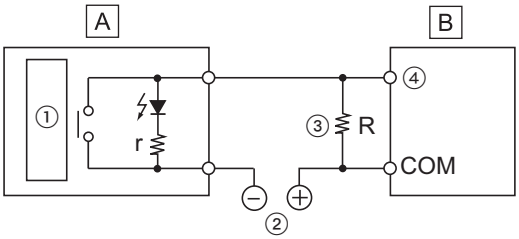
$$W = \frac{(V)^2}{R}$$

V = Power supply voltage

In the actual selection, use a value that is 3 to 5 times the value of W.

Precaution when using an LED-equipped limit switch

If the input of the PLC does not turn off because of the leakage current from the LED-equipped limit switch, the use of a bleeder resistor is recommended, as shown below.



A	LED-equipped limit switch	③	Bleeder resistor
B	FP0R	④	Input terminal
①	Internal circuit	r	Internal resistor of limit switch (kΩ)
②	Power supply for input	R	Bleeder resistor (kΩ)

The OFF voltage of the input is 2.4V. Therefore, when the power supply is 24V, select the bleeder resistor R so that the current will be greater than the result of this formula:

$$I = \frac{24 - 2.4}{r}$$

The resistance R of the bleeder resistor is:

$$R \leq \frac{21.84R}{9.1I - 2.4} \ (k\Omega)$$

The wattage W of the resistor is:

$$W = \frac{(V)^2}{R}$$

V = Power supply voltage

In the actual selection, use a value that is 3 to 5 times the value of W.

6.5.2 Output Wiring

There is no fuse in the output circuit. It is recommended to install external fuses in every circuit, to reduce the risk of burning out the output circuit when the output is shorted.

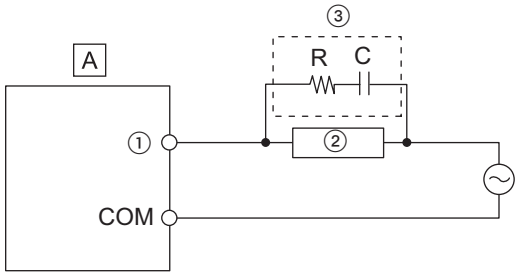
Do not connect a load that exceeds the maximum switching ability of the output terminal.

6.5.2.1 Protective Circuit for Inductive Loads

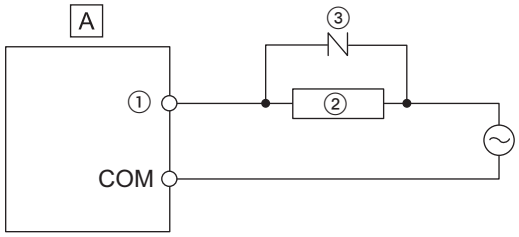
With an inductive load, a protective circuit should be installed in parallel with the load.

When switching DC inductive loads with the relay output type, be sure to connect a diode across the ends of the load.

Using an AC inductive load (relay output type)

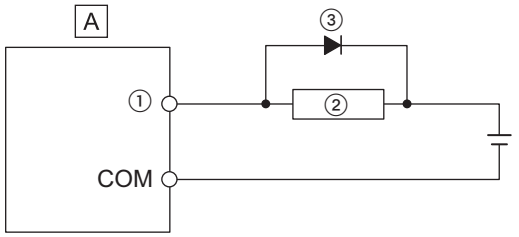


A	FP0R
①	Output terminal
②	Load
③	Surge absorber, e.g. resistance R: 50Ω, capacitance C: 0.47μF



A	FP0R
①	Output terminal
②	Load
③	Varistor

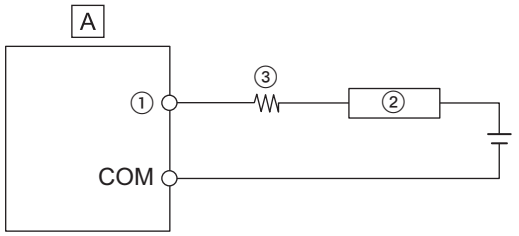
Using a DC inductive load



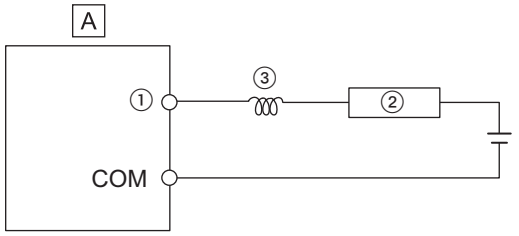
A	FP0R
①	Output terminal
②	Load
③	Diode; reverse voltage (V_R): 3 × load voltage, average rectified forward current (I_0): ≥ load current or more

6.5.2.2 Protective Circuit for Capacitive Loads

When connecting loads with large in-rush currents, connect a protection circuit as shown below to minimize their effect.



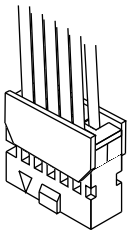
A	FP0R
①	Output terminal
②	Load
③	Resistor



A	FP0R
①	Output terminal
②	Load
③	Inductor

6.6 Wiring the MIL Connector

The connector indicated below is supplied with the unit.



Suitable wire

Size	Cross-sectional area [mm^2]	Insulation thickness [mm]	Rated current
AWG22	0.3	∅ 1.5–1.1mm	3A
AWG24	0.2		

Supplied connector

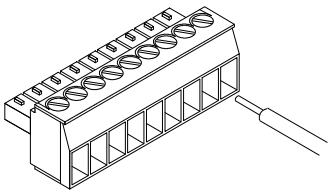
Product no.	Type
AFP0807	10-pin type

Optional cables

Description	Product no.
I/O cable with 10-pin MIL connector, (2pcs: 1 × 10 blue, 1 × 10 white wires), 1m	AFP0521D
I/O cable with 10-pin MIL connector, (2pcs: 1 × 10 blue, 1 × 10 white wires), 3m	AFP0523D
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 blue wires), 1m	AFP0521BLUED
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 blue wires), 3m	AFP0523BLUED
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 colored wires), 1m	AFP0521COLD
I/O cable with 10-pin MIL connector, (2pcs: 2 × 10 colored wires), 3m	AFP0523COLD
I/O cable with 40-pin MIL connector, blue wires, 1m	AYT58403BLUED
I/O cable with 40-pin MIL connector, blue wires, 3m	AYT58406BLUED
I/O cable with 40-pin MIL connector, colored wires according to DIN 47100, 3m	AYT58406COLD

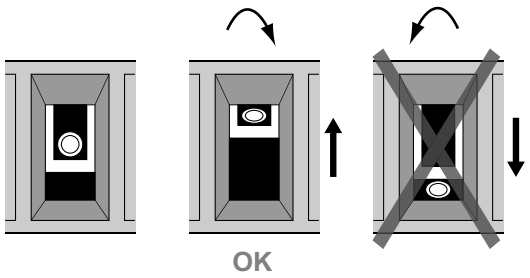
6.7 Wiring the Terminal Block

Screw-type terminal blocks are used. The suitable wires are given below.



◆**NOTE**

- **When removing the wire's insulation, be careful not to scratch the core wire.**
- **Do not twist the wires to connect them.**
- **Do not solder the wires to connect them. The solder may break due to vibration.**
- **After wiring, make sure stress is not applied to the wire.**
- **If the socket in the terminal block closes upon counter-clockwise rotation, the connection is wrong. Disconnect the wire, check the terminal hole, and then re-connect the wire.**



Terminal block

Item	Description
Number of pins	9
Manufacturer	Phoenix Contact Co.
Model	MC1,5/9-ST-3,5
Product no.	1840434

Suitable wire

Size	Cross-sectional area [mm^2]
AWG22	0.3
AWG24–16	0.2–1.25

Pole terminals with compatible insulation sleeve

For pole terminals, please consider the following specifications:

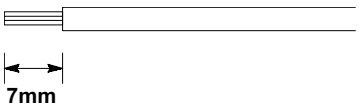
Cross-sectional area [mm^2]	Size
0.25	AWG24
0.50	AWG20
0.75	AWG18
1.00	AWG18
0.5 x 2	AWG20 (for 2 pieces)

The tightening torque should be 0.22–0.25Nm or less. Use a screwdriver with a blade size of 0.4 x 2.5.

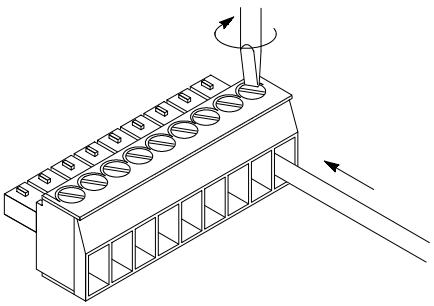
Wiring method

◆**Procedure**

1. **Remove a portion of the wire's insulation**

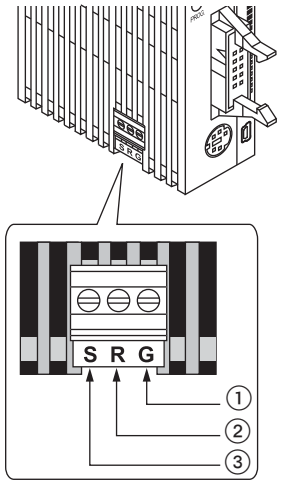


2. **Insert the wire into the terminal block until it contacts the back of the socket**
3. **Turn the screw clockwise to fix the wire in place**



6.8 Wiring the COM Port

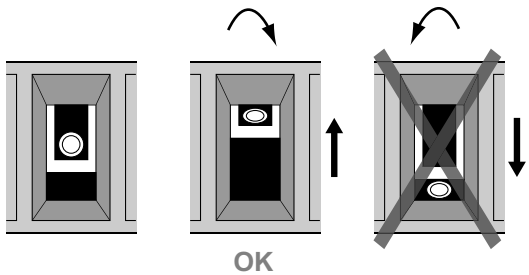
A screw-down connection type is used for the COM port (RS232C port). The suitable wires are given below.



①	Signal Ground
②	Receive Data (Input)
③	Send Data (Output)

◆NOTE

- **When removing the wire's insulation, be careful not to scratch the core wire.**
- **Do not twist the wires to connect them.**
- **Do not solder the wires to connect them. The solder may break due to vibration.**
- **After wiring, make sure stress is not applied to the wire.**
- **If the socket in the terminal block closes upon counter-clockwise rotation, the connection is wrong. Disconnect the wire, check the terminal hole, and then re-connect the wire.**



Terminal block

The communication connector manufactured by Phoenix Contact is used.

Item	Description
Number of pins	3
Manufacturer	Phoenix Contact Co.
Model	MKDS1/3-3.5
Product no.	1751400

Suitable wire

Size	Cross-sectional area [mm^2]
AWG28–16	0.08–1.25

Only use shielded twisted pair cables.

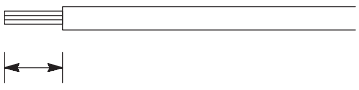
It is recommended to ground the shielded part.

When using a pole terminal, please refer to "Wiring the Terminal Block" on page 76.

Wiring method

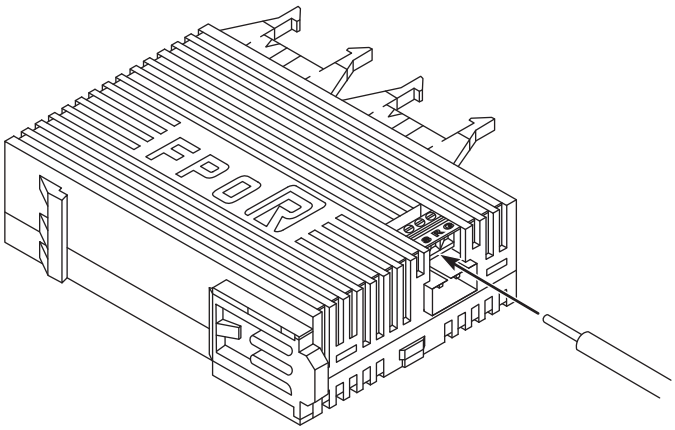
◆**Procedure**

1. **Remove a portion of the wire's insulation**

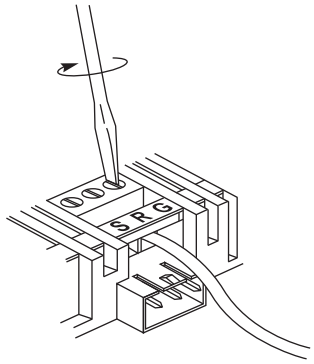


5 mm

2. **Insert the wire into the COM port (RS232C port) until it contacts the back of the socket**



3. Turn the screw clockwise to fix the wire in place



Chapter 7

Communication

7.1 Communication Modes

The FP0R offers four different communication modes:

- MEWTOCOL-COM Master/Slave
- Program controlled
- PLC Link (MEWNET-W0)
- Modbus RTU Master/Slave

Communication ports

The FP0R is equipped with the following ports:

- TOOL port (RS232C interface)
- USB port (USB 2.0 Fullspeed interface)
- COM port (RS232C interface)

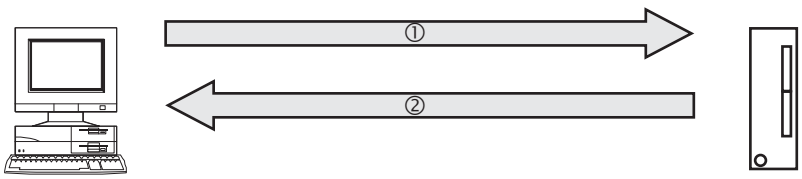
7.1.1 Terminology in FPWIN Pro and FPWIN GR

Although similar, FPWIN Pro and FPWIN GR use slightly different terminology to describe communication modes. The following table provides the terminological equivalents for FPWIN GR.

FPWIN Pro	FPWIN GR
MEWTOCOL-COM Master/Slave	Computer link
Program controlled	General-purpose serial communication
Modbus RTU Master/Slave	MODBUS RTU
PLC Link (MEWNET-W0)	PC (PLC) link

7.1.2 MEWTOCOL-COM Master/Slave

This communication mode uses the proprietary MEWTOCOL-COM protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication. A 1:N network is also known as a C-NET.



MEWTOCOL-COM connection between a computer and the FP0R

①	Command message	②	Response message

There is a MEWTOCOL-COM master function and a MEWTOCOL-COM slave function. The side that issues commands is called master. The slave receives the commands, executes the

process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

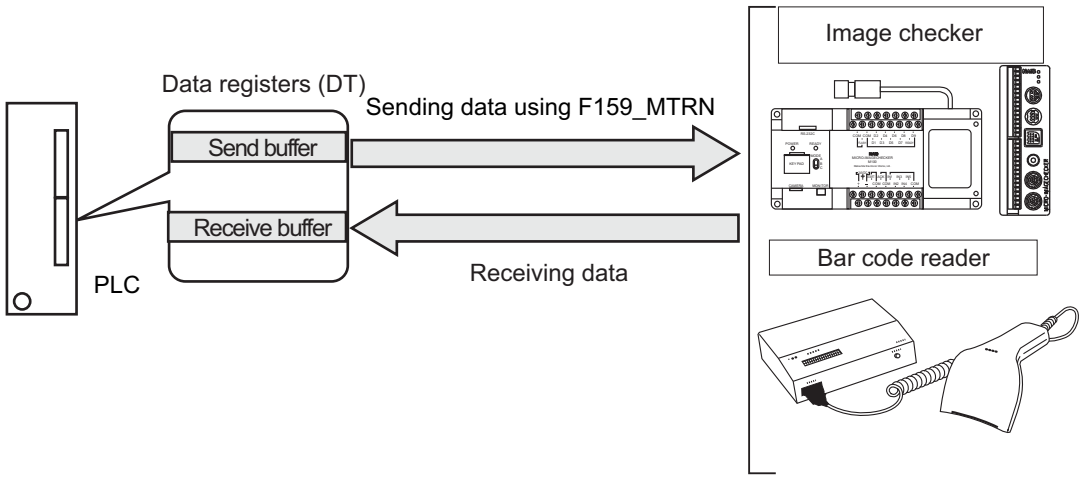
◆REFERENCE

For detailed information on the MEWTOCOL-COM communication mode, see "MEWTOCOL-COM" on page 97.

7.1.3 Program Controlled Communication

With program controlled communication, the user generates a program which governs the data transfer between a PLC and one or more external devices connected to the communication port. By this, any standard or user protocol can be programmed.

Typically, such a user program consists of sending and receiving the data.



Program controlled connection between the FP0R and an external device

◆REFERENCE

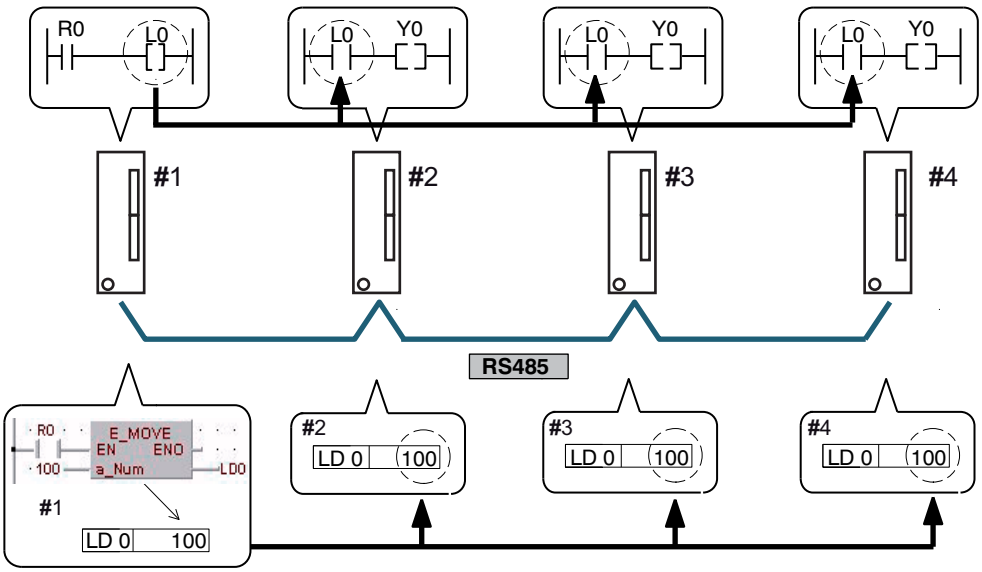
For detailed information on the program controlled communication mode, see "Program Controlled Communication" on page 110.

7.1.4 PLC Link

PLC Link is an economic way of linking PLCs using a twisted-pair cable and the MEWNET protocol. Data is shared with all PLCs by means of dedicated internal relays called link relays (L) and data registers called link registers (LD). The statuses of the link relays and link registers of one PLC are automatically fed back to the other PLCs on the same network. The link relays and link registers of the PLCs contain areas for sending and areas for receiving data. Station numbers and link areas are allocated using the system registers.

◆**EXAMPLE**

Link relay L0 for station #1 turns to TRUE. The status change is fed back to the programs of the other stations, and Y0 of the other stations is set to TRUE. A constant of 100 is written to link register LD0 of station #1. The contents of LD0 in the other stations are also changed to a constant of 100.



PLC Link connection between four FP0R units

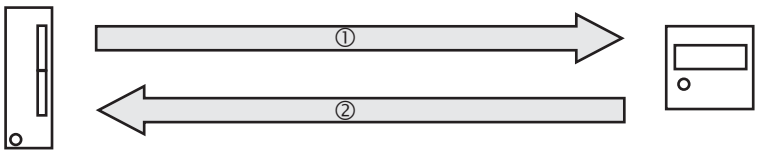
#	Station number of PLC	LD	Link register

◆REFERENCE

For detailed information on the PLC Link communication mode, see "PLC Link" on page 135.

7.1.5 Modbus RTU Master/Slave

This communication mode uses the Modbus RTU protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication.



Modbus RTU connection between the FP0R and an external device

①	Command message	②	Response message

There is a Modbus RTU master function and a Modbus RTU slave function. The side that issues commands is called master. The slave receives the commands, executes the process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

The Modbus protocol supports both ASCII mode and RTU binary mode. However, the PLCs of the FP Series only support the RTU binary mode.

◆REFERENCE

For detailed information on the Modbus RTU communication mode, see "Modbus RTU Communication" on page 152.

7.2 Ports: Names and Principle Applications

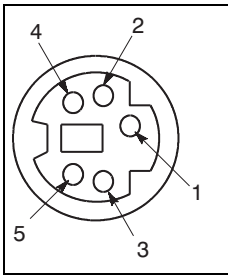
Port name	Connector	Communication mode
TOOL port	Mini DIN 5-pin connector	• MEWTOCOL-COM Slave • Program controlled (in RUN mode only)[1)]
USB port	USB miniB type	• MEWTOCOL-COM Slave
COM port	3-wire RS232C screw type	• MEWTOCOL-COM Master/Slave • Program controlled • Modbus RTU Master/Slave • PLC Link

1) In PROG mode, the TOOL port is automatically set to MEWTOCOL-COM mode even if program controlled mode has been selected. This way it is always possible to communicate in PROG mode with a programming software like FPWIN Pro.

7.2.1 TOOL Port

The TOOL port can be used to connect a programming tool.

A commercial 5-pin mini DIN connector is used for the TOOL port on the CPU.



Pin no.	Signal name	Abbreviation	Signal direction
1	Signal Ground	SG	–
2	Send Data	SD	CPU → External device
3	Receive Data	RD	CPU ← External device
4	(Not used)	–	–
5	+5V	+5V	CPU → External device

The factory settings are shown below. They can be changed in the system registers.

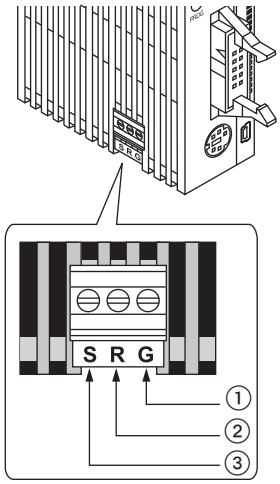
Communication parameter	Factory settings
Baud rate	9600bit/s
Data length	8
Parity	Odd
Stop bit	1bit

Set the station number for the TOOL port in the TOOL port setting area of the system registers.

7.2.2 COM Port

The following CPU types are equipped with a COM port: C10CR, C14CR, C16C, C32C, T32C, F32C

This port is used to connect devices via RS232C to enable data input/output.

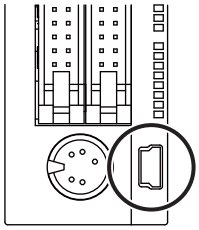


①	Signal Ground
②	Receive Data (Input)
③	Send Data (Output)

7.2.3 USB Port

The USB port can be used to connect a programming tool.

Panasonic's USB cable CABMINIUSB5D or a commercial USB2.0 AB type cable can be used.



To use the USB port, you must install the USB driver (see page 87).

Specifications

Item	Description
Connector	5-pin Mini-B type
Standard (baud rate)	USB2.0 Fullspeed
Communication mode	MEWTOCOL-COM Slave

◆CAUTION

Install the programming tool before connecting the FP0R with a PC.

If you connect the FP0R to a PC with the USB cable before the programming tool is installed or during installation, the USB driver will not be installed correctly.

USB port settings

The settings for the USB port are fixed and cannot be changed.

Connecting the PLCs with a personal computer using a USB cable enables communication with our programming software.

This communication method uses the USB as a virtual serial port, i.e. the FP0R connected via USB is treated by the PC as if connected via the COM port. The COM port number of the COM port allocated for the USB is fixed unless you change the number.

You need only perform the connection procedure the first time you establish the USB connection.

However, you must change the communication setting when switching between the USB and TOOL port connection.

System requirements

- Operating system on the PC:
  - Windows®2000
  - Windows®XP
  - Windows®Vista
- FPWIN Pro version 6.1 or later, or FPWIN GR version 2.80 or later
- USB cable (see page 13)

◆NOTE

- **A USB hub cannot be used.**
- **When multiple FP0R units are connected to one PC with the USB, they cannot communicate with the PC simultaneously. The PC can communicate with the FP0R that was connected first only, and it cannot communicate with the other FP0R.**

7.2.3.1 Installing the USB Driver

The following two USB drivers must be installed to recognize the USB:

- USB driver
- USB-COM conversion driver

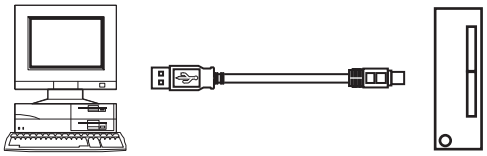
The installation procedure may differ depending on the PC's OS.

◆NOTE

For a PC with more than one connector, you may be requested to reinstall these two drivers if the USB connectors' positions have changed.

◆Procedure

1. **Turn on the power supply of the FP0R**
2. **Connect the FP0R with a PC using a USB cable**



The PC recognizes the USB driver automatically.

3. **Follow the wizard's instructions**

Confirming COM Ports

The FP0R connected to the PC via USB is treated as if connected via a COM port. It depends on your PC environment to which COM port the USB is allocated. Therefore, it is necessary to confirm the COM port number.

A COM port number is necessary for communication with the programming tool.

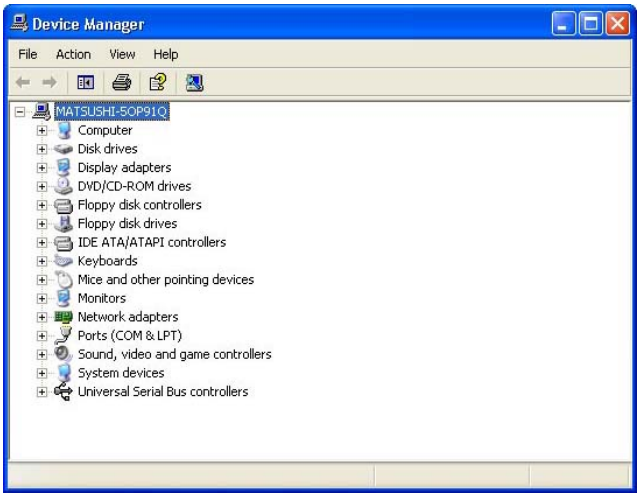
◆Procedure

1. **Display Device Manager**

   For **Windows®XP**: My computer → View system information →Hardware tab → Device Manager.

   For **Windows®2000**: My computer → Control panel → System →Hardware tab → Device manager → View → Device by type.

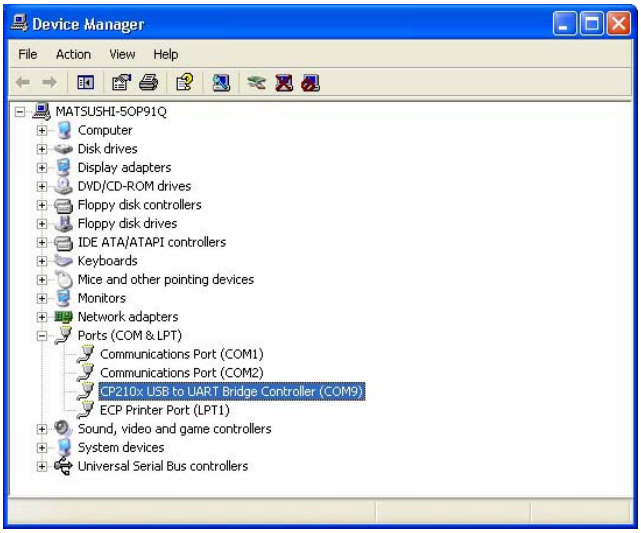
For **Windows®98 Second Edition/Windows®Me**: My computer →Control panel →System →Device manager tab → View devices by type.



2. **Double-click "Ports (COM & LPT)"**

3. **Confirm the COM port no.**

   "CP210x USB to UART Bridge Controller (COM n)" is the COM port allocated. COM9 is allocated in the following display.



◆**NOTE**

If "? CP210x USB to UART Bridge Controller" appears in "Other devices" or "Unknown device" is indicated, the installation has failed. Reinstall the USB driver (see page 91).

7.2.3.2 Communication with the Programming Tool

In FPWIN Pro, follow these steps:

◆Procedure

1. **Online → Communication Parameters**
2. **Make the following settings in the "Communication Setting" dialog:**

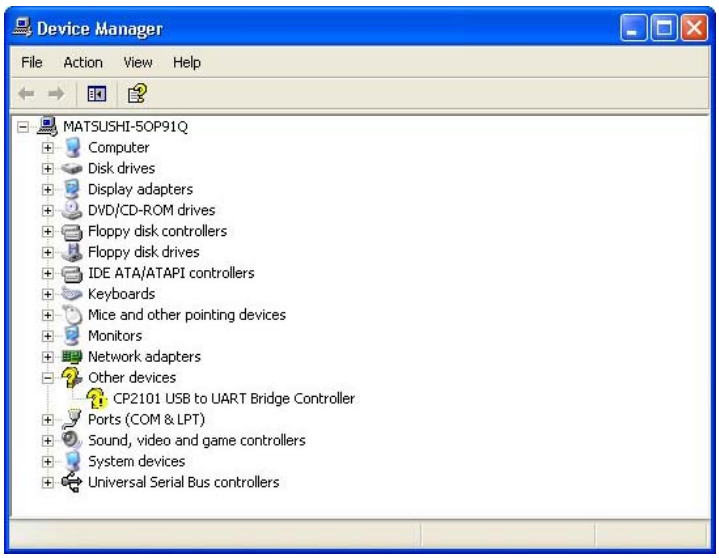
Parameter	Setting
Network type	C-NET (RS232C)
COM port	COM port number allocated for the USB
Baud rate	115200bit/s (Communicates with 115200bit/s when the USB is connected)
Data length	8 bits
Stop bit	1 bit
Parity	Odd

◆REFERENCE

Please refer to your programming tool's online help for information on the COM port settings.

7.2.3.3 Reinstalling the USB Driver

The USB driver must be installed again if the installation failed. If "? CP210x USB to UART Bridge Controller" appears in "Other devices" or "Unknown device" is indicated, the installation has failed.



Also, reinstall the driver if the USB connection does not work well.

Reinstalling the USB driver

◆**Procedure**

1. **Right-click "? CP210X USB to UART Bridge Controller"**
2. **Select "Delete"**
3. **Reinstall the USB driver (see page 89)**

7.3 Communication Specifications

TOOL port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Slave Modem initialization Program controlled (in RUN mode only)

USB port

Item	Description
Standard (baud rate)	USB2.0 Fullspeed
Communication mode	MEWTOCOL-COM Slave

COM port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Master/Slave Modem initialization Program controlled Modbus RTU Master/Slave PLC Link

Default settings

Port	Baud rate	Data length	Parity	Stop bit
TOOL port	9600bit/s	8 bits	Odd	1 bit
COM port	9600bit/s	8 bits	Odd	1 bit

7.4 Communication Parameters

The communication parameters are set in the system registers of the PLC. Make settings for the communication mode, communication format, baud rate, station number, and receive buffer if necessary.

During PROG mode:

Use the programming tool to enter settings for the communication port in the system registers.

During RUN mode:

Use the SYS1 instruction to change the communication parameters. Please refer to the FPWIN Pro online help for detailed information.

The communication mode can be switched using F159_MRTN (see page 96).

7.4.1 Setting System Registers in PROG Mode

◆Procedure

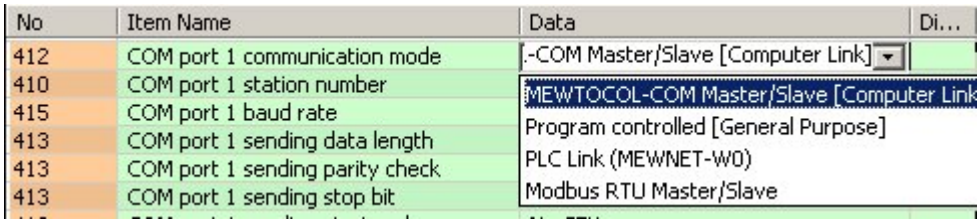
1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **Double-click "COM Port"**

   To make settings for the TOOL port, select "TOOL Port" under "System Registers".

The following communication parameters are set in the system registers:

Communication mode

Select a communication mode. The factory setting for the communication mode is "MEWTOCOL-COM Master/Slave".



Station number

The station number must be set for MEWTOCOL-COM Master/Slave, Modbus RTU, and for PLC Link.

MEWTOCOL-COM **Modbus RTU**	The station number can be set within a range of 1 to 99. In the FP0 compatibility mode, the station number can be set within a range of 1 to 32.
PLC Link	The station number can be set within a range of 1 to 16.

By default, the station number for each communication port is set to 1 in the system register settings. There is no need to change this for 1:1 communication, but if 1:N communication is used to connect multiple PLCs to the transmission line, the station number must be specified to identify the different PLCs.

The station number is specified either by using

A. the SYS1 instruction

B. system register settings in the programming tool

The priority for setting the station number is in the above order.

◆REFERENCE

For details on the SYS1 instruction, please refer to your Programming Manual or to the FPWIN Pro online help.

Baud rate

The default baud rate for the various ports is 9600bit/s. Select a value from 2400 to 115200bit/s.

Lower baud rates of 300, 600, and 1200bit/s can be specified using the SYS1 instruction. However, this will not change the setting value of the system register.

The setting must match the external device connected to the communication port.

PLC Link: The baud rate is fixed at 115200bit/s.

FP0 compatibility mode:

TOOL port	9600 or 19200bit/s
COM port	300, 600, 1200, 2400, 4800, 9600, or 19200bit/s

Communication format setting

Default settings:

Data length:	8 bits
Parity:	Odd
Stop bit:	1 bit
End code:	CR
Start code:	No STX

The setting must match the external device connected to the communication port.

MEWTOCOL-COM Modbus RTU	The end code setting must always be "CR", and the start code setting must be "No STX".
PLC Link	The communication format settings are fixed.

Receive buffer

For program controlled communication, a receive buffer must be specified in the system registers. Set a value for receive buffer starting address and receive buffer capacity. See "Setting Communication Parameters" on page 137.

7.4.2 Changing Communication Mode in RUN Mode

The communication mode of the CPU's communication ports can be changed during RUN mode. You can toggle between program controlled mode and MEWTOCOL-COM mode by executing F159_MTRN and setting the variable **n_Number** (the number of bytes to be sent) to 16#8000.

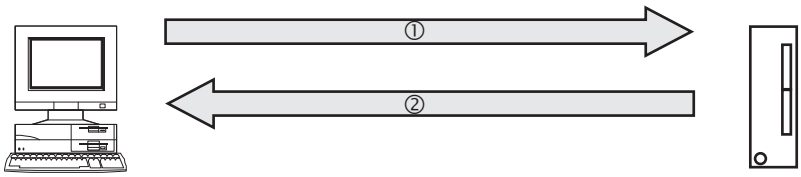
Please refer to the FPWIN Pro online help on F159_MTRN for a programming example.

◆**NOTE**

- **When the power is turned on, the communication mode selected in the system registers is set.**
- **It is not possible to change to the Modbus RTU mode using F159_MTRN.**

7.5 MEWTOCOL-COM

This communication mode uses the proprietary MEWTOCOL-COM protocol to exchange data between a master and one or more slaves. This is called 1:1 or 1:N communication. A 1:N network is also known as a C-NET.



①	Command message	②	Response message

MEWTOCOL-COM connection between a computer and the FP0R

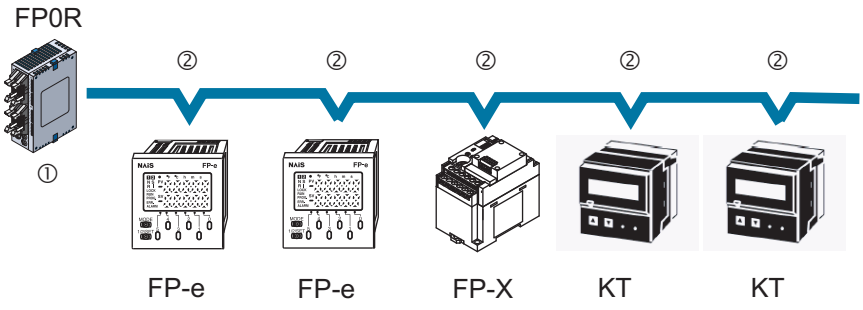
There is a MEWTOCOL-COM master function and a MEWTOCOL-COM slave function. The side that issues commands is called master. The slave receives the commands, executes the process and sends back responses. The slave answers automatically to the commands received from the master, so no program is necessary on the slave.

MEWTOCOL-COM master function

The master can be a PLC or any external device supporting the master function. To use the built-in master functionality of the PLC, select MEWTOCOL-COM Master/Slave in the system registers and implement a PLC program. The applicable instructions are F145_WRITE_DATA and F146_READ_DATA.

MEWTOCOL-COM Master/Slave mode is recommended over program controlled mode since programming is easier.

The master function can be used for communication with all Panasonic devices equipped with a MEWTOCOL-COM slave function, for example PLCs, Imagecheckers, temperature controllers, or eco-power meters.



①	Master	②	Slave

◆NOTE

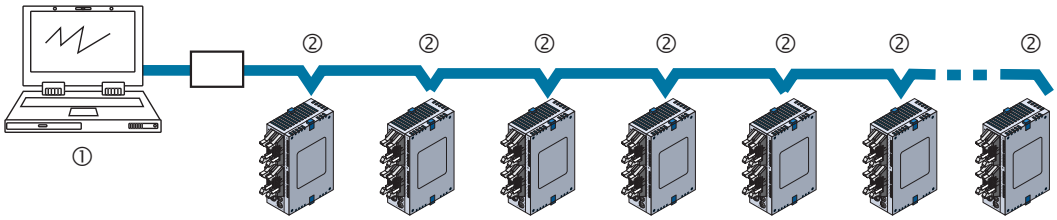
- **The master function cannot be used with the TOOL port or with the USB port.**

- **Do not execute the F145_WRITE_DATA and F146_READ_DATA instructions when the unit is used as a slave unit.**

MEWTOCOL-COM slave function

The slave can be a PLC or any external device which supports the MEWTOCOL-COM protocol. The slave automatically receives a command, processes it and sends back a response. To use the built-in slave functionality of the PLC, select "MEWTOCOL-COM Master/Slave" in the system registers. For 1:N communication in a C-NET, the station number must be specified in the system registers of the slave. No program is necessary on the slaves.

The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.



①	Master	②	Slave

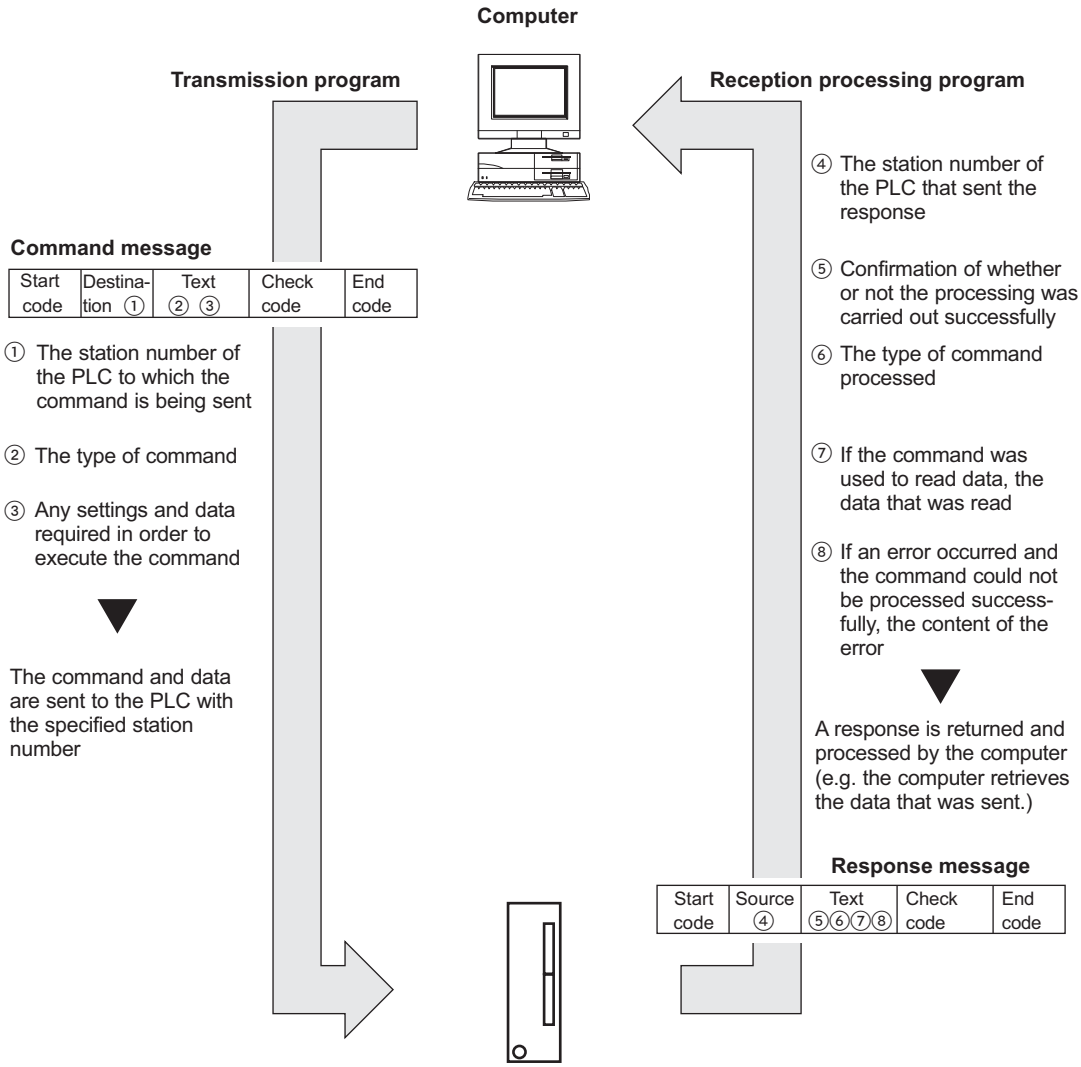
◆NOTE

Panasonic offers software tools with implemented MEWTOCOL-COM Master functionality:

- **Control FP Connect – connects your Visual Basic application to Panasonic PLCs**
- **PCWAY – displays PLC data in Excel**

7.5.1 Operation Outline for MEWTOCOL-COM Slave

Instructions issued by the computer to the PLC are called commands. Messages sent back to the computer from the PLC are called responses. When the PLC receives a command, it processes the command regardless of the sequence program, and sends a response back to the computer. Communication is carried out in a conversational format, based on the MEWTOCOL-COM communication procedures. Data is sent in ASCII format. The computer has the first right of transmission. The right of transmission shifts back and forth between the computer and the PLC each time a message is sent.

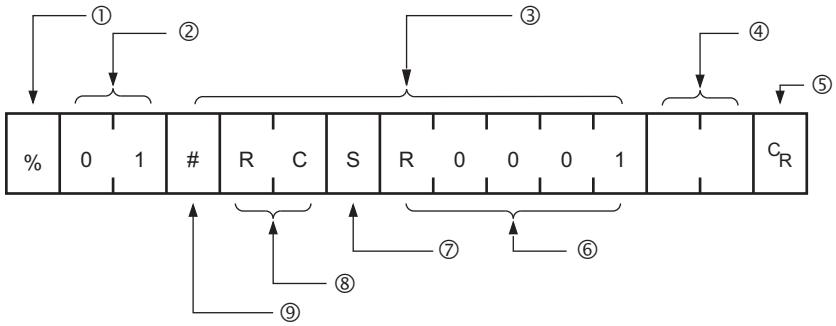

Computer
Transmission program
Reception processing program
Command message
Start code
Destination ①
Text ② ③
Check code
End code
① The station number of the PLC to which the command is being sent
② The type of command
③ Any settings and data required in order to execute the command
The command and data are sent to the PLC with the specified station number
④ The station number of the PLC that sent the response
⑤ Confirmation of whether or not the processing was carried out successfully
⑥ The type of command processed
⑦ If the command was used to read data, the data that was read
⑧ If an error occurred and the command could not be processed successfully, the content of the error
A response is returned and processed by the computer (e.g. the computer retrieves the data that was sent.)
Response message
Start code
Source ④
Text ⑤⑥⑦⑧
Check code
End code

MEWTOCOL-COM communication between the FP0R and a computer

7.5.2 Command and Response Format

Command message

All command-related items should be noted in the text segment. The station number must be specified before sending the command.



① **Start code**

Commands must always have a "%" (ASCII code: 16#25) or a "<" (ASCII code: 16#3C) at the beginning of a message.

The FP0R supports an expansion start code ("<") to send single frames of up to 2048 characters. Using the start code "%", a maximum of 118 characters can be sent in one frame.

② **Station number**

The station number of the slave to which you want to send the command must be specified.

The range is 01 to 99 (decimal).

In 1:1 communication, the station number "01" (ASCII code: 16#3031) should be specified.

③ **Text**

The content differs depending on the command. The content should be noted in all upper-case characters, following the fixed formula for the particular command.

The method for writing text segments in the message varies depending on the type of command.

④ **Check code**

Hexadecimal block check code (BCC) for error detection using horizontal parity. The BCC should be created so that it targets all of the text data from the header to the last text character.

The BCC starts from the header and checks each character in sequence, using the exclusive OR operation, and replaces the final result with character text. It is normally part of the calculation program and is created automatically.

The parity check can be skipped by entering "* *" (ASCII code: 16#2A2A) instead of the BCC.

⑤ **End code**

Messages must always end with a "$^{C}_{R}$" (ASCII code: 16#0D).

⑥ **Target address**

Address of the target area to be read or written (e.g. internal relay R1)

⑦ **Data area**

Specification of the number of points to be read or written (S = 1 point)

⑧ **Command name**

e.g. RC, read contact area

⑨ **Command code**

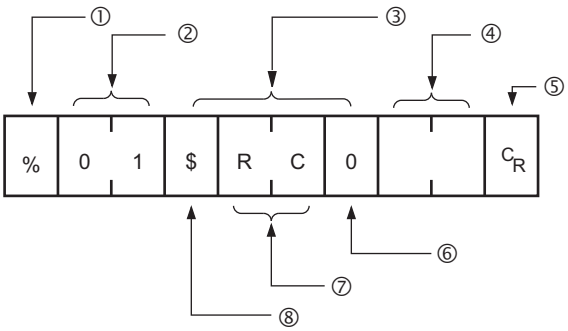
(16#23) indicates that this is a command

◆**NOTE**

If there are large numbers of characters to be written, they may be divided and sent as several commands. If there is a large number of characters in the value that was loaded, they may be divided and several responses sent.

Response message

The slave that received the command in the example above sends the processing results to the computer.



① **Start code**

A "%" (ASCII code: 16#25) or "<" (ASCII code: 16#3C) must be at the beginning of a message. The response must start with the same start code as the command.

② **Station number**

The station number of the slave that processed the command is stored here.

③ **Text**

The content of this varies depending on the type of command. The value should be read based on the content. If the processing is not completed successfully, an error code will be stored here, so that the content of the error can be checked.

④ **Check code**

Hexadecimal block check code (BCC) for error detection using horizontal parity. The BCC starts from the header and checks each character in sequence, using the exclusive OR operation, and replaces the final result with character text.

⑤ **End code**

There is always a "$^{C}_{R}$" (ASCII code: 16#0D) at the end of the message.

⑥ **Data**

For a read command, the data read is stored here.

⑦ **Command name/error code**

Normal processing: The command name is stored here.

Error condition: The error code is stored here.

⑧ **Response code**

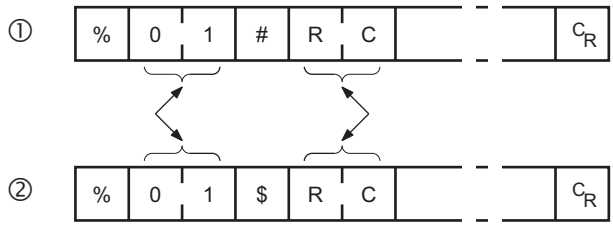
Normal processing: "$" (ASCII code: 16#24)

Error condition: ! (ASCII code: 16#21)

If the response contains an "!" instead of a "$", check the meaning of the error code.

◆**NOTE**

- **If no response is returned, the command may not have arrived at the slave, or the slave may not be functioning. Check to make sure all of the communication specifications (e.g. baud rate, data length, and parity) match between the master and the slave.**
- **Station number and command name are always identical in a command and its corresponding response (see below). This makes the correspondence between a command and a response clear.**



①	Command message	②	Response message

7.5.3 Commands

Command name	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the on and off status of contacts. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts on and off. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the value set for a timer/counter.
Write timer/counter set value area	WS	Writes a timer/counter setting value.
Read timer/counter elapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using MD and MC.
Preset contact area (fill command)	SC	Embeds the area of a specified range in a 16-point on and off pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads the specifications of the PLC and error codes if an error occurs.

Command name	Code	Description
Remote control	RM	Switches the operation mode of the PLC.
Abort	AB	Aborts communication.

7.5.4 Setting Communication Parameters

Make the following settings for the communication port:

- communication mode (MEWTOCOL-COM)
- station number
- baud rate
- communication format

For details on setting the communication parameters, see "Setting System Registers in PROG Mode" on page 94.

◆NOTE

- **The end code setting must always be "CR", and the start code setting must be "No STX".**
- **The station number can be set within a range of 1 to 99.**
- **With a C-NET adapter, a maximum of 32 stations can be specified.**

7.5.4.1 FP0 Compatibility Mode

Make sure that the PLC type selected in FPWIN Pro is "FP0".

All ports can be used in FP0 compatibility mode. For the USB port, the settings are fixed.

Make the following settings for the communication port:

TOOL port

- station number
- modem connection (disable/enable)
- communication format (sending data length)
- baud rate

COM port

- communication mode
- station number
- baud rate
- communication format
- modem connection (disable/enable)

For details on setting the communication parameters, see page 94.

◆**NOTE**

- **The end code setting must always be "CR", and the start code setting must be "No STX".**

7.5.5 1:1 Slave Communication

System register settings

For 1:1 MEWTOCOL-COM communication, the system registers should be set as shown below.

No.	Name	Set value
410	COM port 1 - station number	1
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave
413	COM port 1 - communication format	Data length: 8 bits Parity: Odd Stop bit: 1 bit End code: CR Start code: No STX
415	COM port 1 - baud rate	2400–115200bit/s

◆**NOTE**

The communication format and baud rate of the PLC should be set to match the connected device.

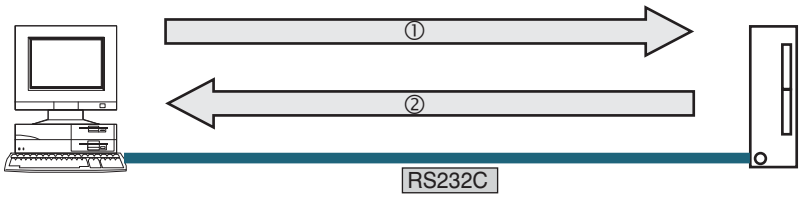
Programming

For MEWTOCOL-COM communication, a program must be created that allows command messages to be sent and response messages to be received on the computer side. There is no programming required on the slave. Only the station number and the communication parameters must be set in the system registers. The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.

If a software program such as PCWAY is used on the computer side, PLC data can easily be read and written without having to think about the MEWTOCOL-COM protocol.

7.5.5.1 1:1 Communication with a Computer

For a 1:1 MEWTOCOL-COM connection between the FP0R and a computer, an RS232C cable is needed. Communication is performed via commands from the computer and responses from the PLC.

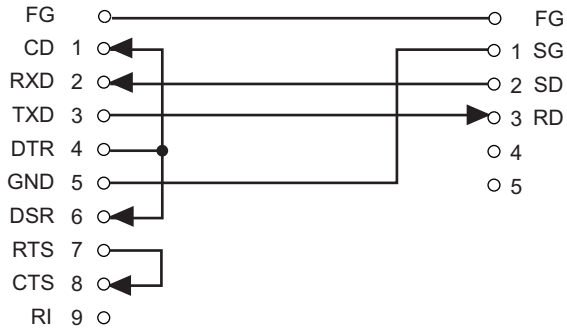


①	Command message	②	Response message

1:1 MEWTOCOL-COM connection between a computer and the FP0R

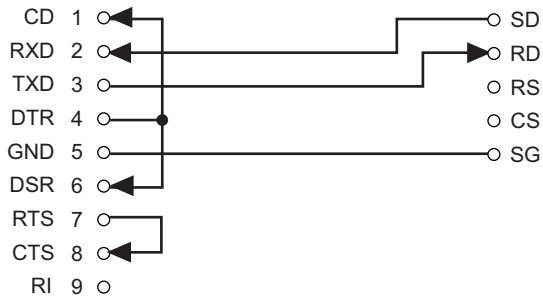
It is recommended to connect the computer to the TOOL port of the FP0R. A connection cable (order no. AFC8513D) with a 5-pin mini-DIN connector and a 9-pin Sub-D connector is available.

- Using the TOOL port



Left: computer, right: FP0R

- Using the COM port



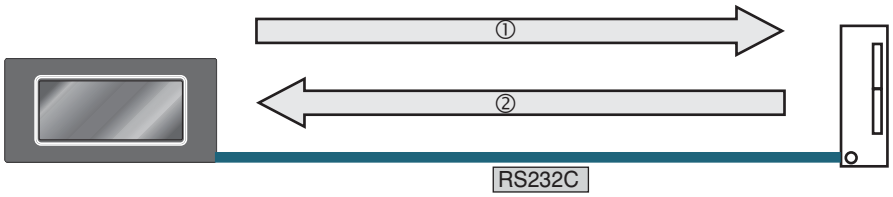
Left: computer, right: FP0R

7.5.5.2 1:1 Communication with GT Series Programmable Displays

For a 1:1 MEWTOCOL-COM connection between the FP0R and a programmable display of the GT series, an RS232C cable is needed. Communication is performed via commands from the programmable display and responses from the PLC.

No program is required for communication. Simply set the mutual communication settings to operate the PLC via the programmable display.

It is recommended to connect the computer to the TOOL port of the FP0R. A connection cable (order no. AFC8513D) with a 5-pin mini-DIN connector and a 9-pin Sub-D connector is available.



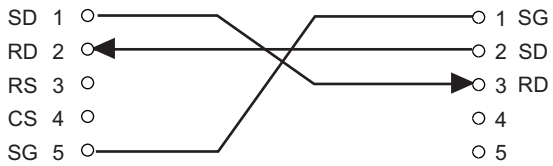
①	Command message	②	Response message

MEWTOCOL-COM connection between a programmable display of the GT series and the FP0R

◆**NOTE**

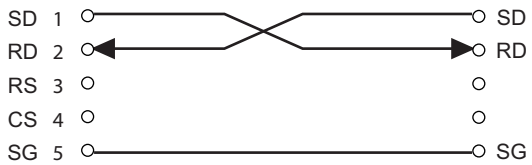
A USB cable cannot be used.

- Using the TOOL port



Left: GT terminal, right: FP0R

- Using the COM port



Left: GT terminal, right: FP0R

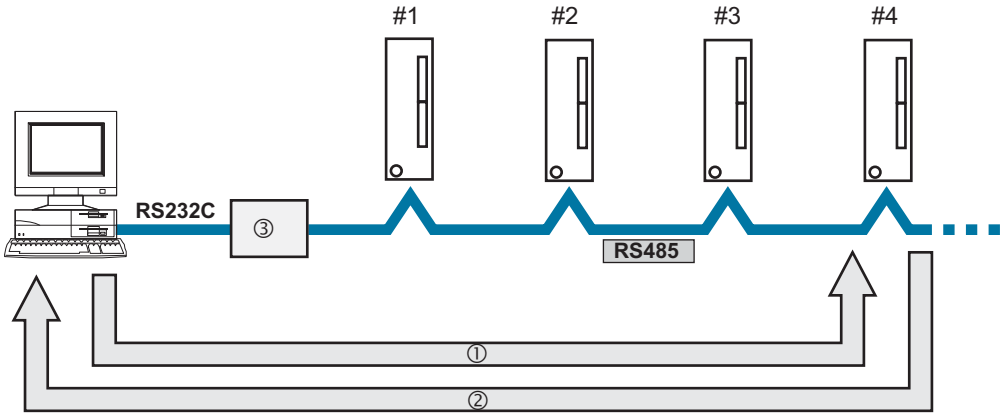
◆REFERENCE

Please refer to the "GT Series Technical Manual" for more information.

7.5.6 1:N Slave Communication

For a 1:N MEWTOCOL-COM connection between a computer and several PLCs, the computer and the first PLC are connected through a commercially available RS232C-RS485 converter. The other PLCs are wired using an RS485 cable.

The computer and the PLCs communicate via commands and responses: The computer sends a command specifying the station number, and the PLC with that station number sends a response back to the computer.



1:N communication between a computer and several PLCs

①	The station number of the PLC to which the command is being sent is included in the command message.
②	The station number of the PLC sending a response is included in the response message.
③	Commercially available converter
#	Station number of PLC

System register settings

For 1:N MEWTOCOL-COM communication, the system registers for COM port 1 should be set as shown below.

No.	Name	Set value
410	COM port 1 - station number	1 to 99 (with C-NET adapter, a maximum of 32 stations is possible)
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave
413	COM port 1 - communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR Start code: No STX
415	COM port 1 - baud rate	2400–115200bit/s

◆NOTE

- **The communication format and baud rate of the PLC should be set to match the connected device.**

- **Lower baud rates of 300, 600, and 1200bit/s can be specified using the SYS1 instruction. However, this will not change the setting value of the system register.**

Programming

There is no programming required on the slave. Only the station number and the communication parameters must be set in the system registers. The program for the master side must send and receive commands according to the MEWTOCOL-COM protocol. MEWTOCOL-COM contains the commands used to control and monitor the slave operation.

If a software program such as PCWAY is used on the computer side, PLC data can easily be read and written without having to think about the MEWTOCOL-COM protocol.

7.5.7 Sample Program for Master Communication

Use the F145_WRITE and F146_READ instructions for the MEWTOCOL-COM master function. Be sure to set the COM port used in the program to "MEWTOCOL-COM Master/Slave" in the system registers.

GVL

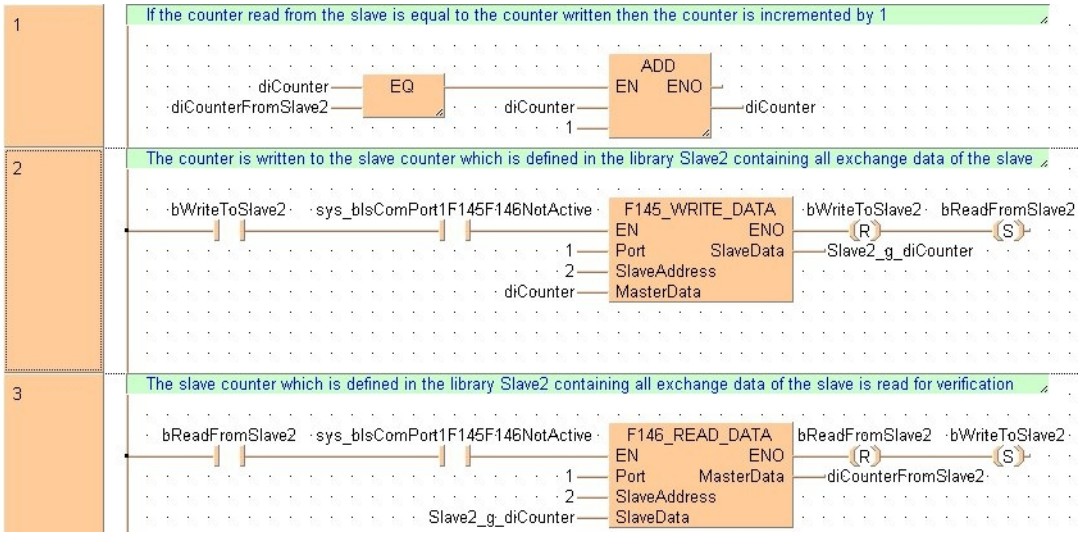
	Class	Identifier	FP Address	IEC Address	Type
0	VAR_GLOBAL	Slave2_g_diCounter	DDT100	%MD5.100	DINT

POU Header

	Class	Identifier	Type	Initial
0	VAR_EXTERNAL	Slave2_g_diCounter	DINT	0
1	VAR	diCounter	DINT	0
2	VAR	diCounterFromSlave2	DINT	-1
3	VAR	bWriteToSlave2	BOOL	TRUE
4	VAR	bReadFromSlave2	BOOL	FALSE

In order to have consistent data in the master project and in the slave project, the common data should be kept in the GVL of a common library.

LD Body



ST Body

```
(* If the counter read from the slave is equal to the counter written
   then the counter is incremented by 1 *)
if (diCounter=diCounterFromSlave2) then
    diCounter:=diCounter+1;
end_if;

if (bWriteToSlave2 AND sys_bIsComPort1F145F146NotActive) then
    (* The counter is written to the slave counter
       which is defined in the library Slave2 containing all exchange data of the slave *)
    F145_WRITE_DATA(Port := 1,
                    SlaveAddress := 2,
                    MasterData := diCounter,
                    SlaveData => Slave2_g_diCounter;
    bWriteToSlave2:=FALSE;
    bReadFromSlave2:=TRUE;
end_if;

if (bReadFromSlave2 AND sys_bIsComPort1F145F146NotActive) then
    (* The slave counter
       which is defined in the library Slave2 containing all exchange data of the slave
       is read for verification *)
    F146_READ_DATA(Port := 1,
              SlaveAddress := 2,
              SlaveData := Slave2_g_diCounter,
              MasterData=> diCounterFromSlave2);
    bReadFromSlave2:=FALSE;
    bWriteToSlave2:=TRUE;
end_if;
```

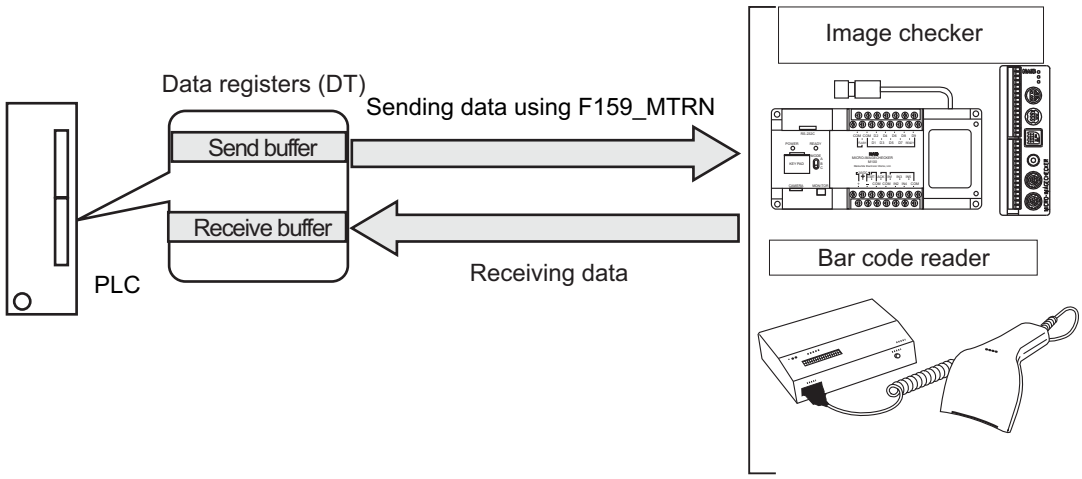
◆REFERENCE

Please refer to the FPWIN Pro online help for detailed information.

7.6 Program Controlled Communication

With program controlled communication, the user generates a program which governs the data transfer between a PLC and one or more external devices connected to the communication port. By this, any standard or user protocol can be programmed.

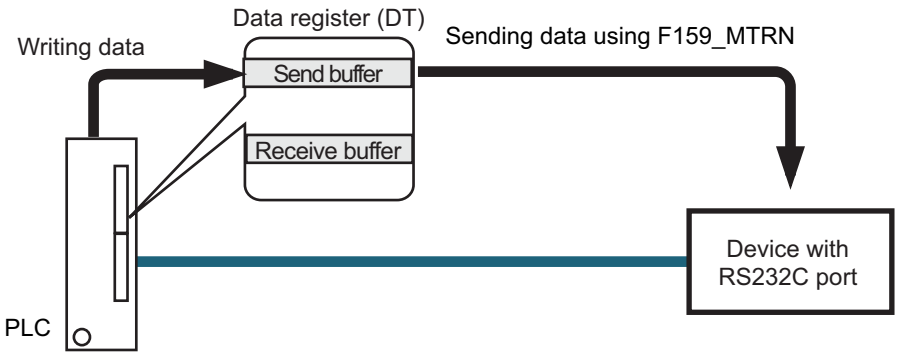
Typically, such a user program consists of sending and receiving the data. The data to be sent and the data received are stored in data register areas defined as send and receive buffers.



Program controlled connection between the FP0R and an external device

Sending data

Sending includes generating the data for the send buffer and sending it using the instruction F159_MTRN. (See also "Sending Data to External Devices" on page 113.) Sending can be controlled by the "transmission done" flag. (See also "Flag Operation" on page 121.)

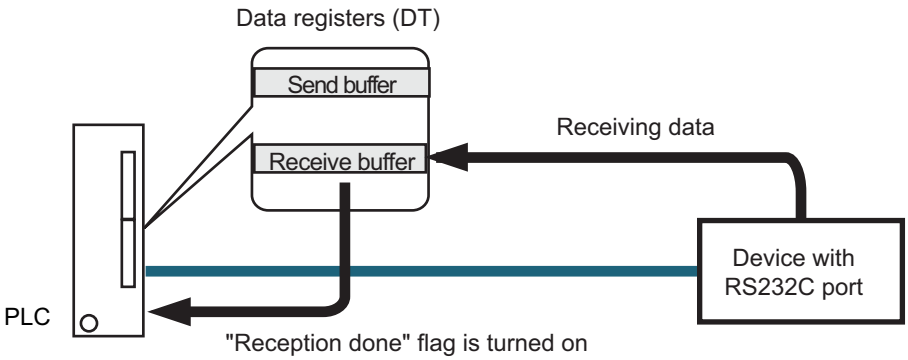


The start and end codes specified in the system registers are automatically added to the data sent. The maximum volume of data that can be sent is 2048 bytes.

Receiving data

Data is automatically received in the receive buffer defined in the system registers. Receiving includes processing the data in the receive buffer and preparing the system to receive further

data. (See also "Receiving Data from External Devices" on page 117.) Reception can be controlled by the "reception done" flag or by directly evaluating the receive buffer. (See also "Flag Operation" on page 121.)



No end code is included in the data stored. The maximum volume of data that can be received is 4094 bytes.

◆NOTE

In the FP0 compatibility mode, F159_MTRN is automatically translated into F144_TRNS.

7.6.1 Setting Communication Parameters

Make the following settings for the communication port:

- communication mode (Program controlled)
- baud rate
- communication format
- receive buffer

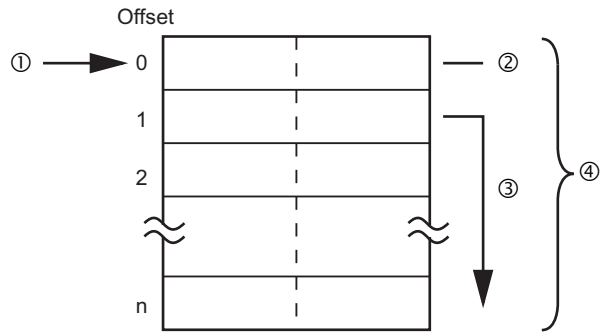
For details on setting the communication parameters, see "Setting System Registers in PROG Mode" on page 94.

◆NOTE

Program controlled mode is also available via the TOOL port.

Specifying a receive buffer

For program controlled communication, a receive buffer must be specified in the DT memory area. The maximum area is 2048 words. Specify the starting address and the capacity (number of words). The receive buffer layout is shown below.



①	Starting address
②	Storage area for the number of bytes received
③	Storage area for the data received
④	Capacity

The receive buffer is specified in the system registers (see page 94):

413	COM port 1 sending end code/reception done condition	CR	CR	Sele
417	COM port 1 receive buffer starting address	200	0 to 1657	The
418	COM port 1 receive buffer capacity	9	0 to 1658	DT2
416	COM port 1 modem connection	Disable	Disable	Spec

◆**NOTE**

FPWIN Pro: In order to use the data in the receive buffer, define a global variable having the same starting address and capacity.

The setting range for the receive buffer starting address is different for the 16k and the 32k type.

7.6.1.1 FP0 Compatibility Mode

Make sure that the PLC type selected in FPWIN Pro is "FP0".

In the FP0 compatibility mode, only the COM port can be used.

Make the following settings for the communication port:

COM port

- communication mode
- station number
- baud rate
- communication format
- receive buffer starting address
- receive buffer capacity

Please be aware that the setting ranges of the FP0 apply if the FP0R is used in FP0 compatibility mode.

For details on setting the communication parameters, see page 94.

◆**NOTE**

The end code setting must always be "CR", and the start code setting must be "No STX".

7.6.2 Sending Data to External Devices

Steps for sending data to external devices:

1. Setting the communication parameters to match the external device
2. Generating the data in the send buffer
3. Sending the data using the instruction F159_MTRN

◆**NOTE**

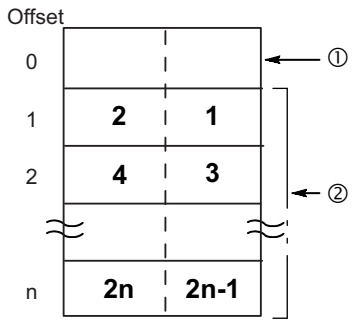
F159_MTRN allows multiple communication ports to be accommodated. This instruction is an updated version of F144_TRNS. Both instructions are compatible with all PLCs: PLCs with multiple communication ports will compile F144_TRNS s, n to F159_MTRN s_Start, n_Number, d_Port*=1. PLCs with only one communication port will compile F159_MTRN to F144_TRNS s, n.

1. Setting the communication parameters (see "Setting Communication Parameters" on page 111)

2. Generating the data in the send buffer

To generate the data in the send buffer, define a variable in the program and copy the data to the send buffer using a transfer instruction, e.g. F10_BKMV.

The storage area for the data to be sent starts with the second word of the send buffer (offset 1). Offset 0 contains the number of bytes to be sent.



① Storage area for the number of bytes to be sent

② Storage area for the data to be sent

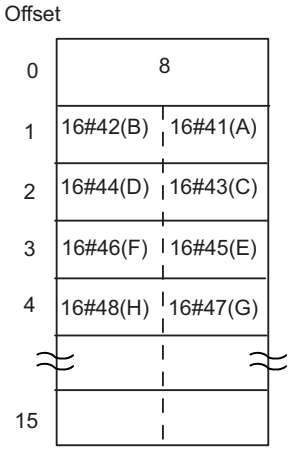
Bold numbers indicate the order of transmission.

The maximum volume of data that can be sent is 2048 bytes.

◆EXAMPLE

Define a send buffer for 30 bytes (ARRAY [0...15] OF WORD) and copy 8 characters of a string ("ABCDEFGH") into the buffer.

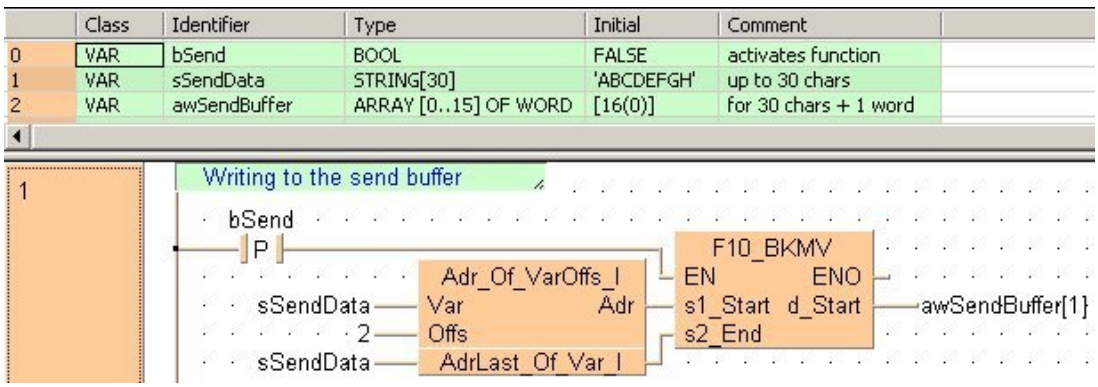
Send buffer layout:



The first word of the send buffer (offset 0) is reserved for the number of bytes to be sent. Therefore, copy the data into offset 1 (**SendBuffer[1]**).

When sending begins (the execution condition for F159_MTRN turns to TRUE), the value in offset 0 is set to 8. At the end of transmission, the value in offset 0 is automatically reset to 0. The data in offset 1 to offset 4 is sent in order from the low order byte.

POU Header and LD Body



ST Body

```
if (DF(bSend)) then
    (* Creating the send buffer *)
    F10_BKMV(s1_Start := Adr_Of_VarOffs(Var := sSendData, Offs := 2),
    s2_End := AdrLast_Of_Var(sSendData), d_Start => awSendBuffer[1]);
```

When the variable **bSend** is set to TRUE, the function F10_BKMV copies the characters of the string **sSendData** to the buffer **awSendBuffer** beginning at **awSendBuffer[1]**.

The first two words of a string contain the string header information (maximum number of characters and the current number of characters). The string header must not be copied into the buffer. Therefore, enter an offset of 2 to the starting address of the string before copying the data.

Make sure that the send buffer is big enough for all the data to be sent. To determine its size you must take into account that two characters of the string **SendString** can be copied into each element of the array **SendBuffer**. **SendBuffer[0]** is reserved for the total number of bytes to be sent by F159_MTRN.

3. Sending the data using the instruction F159_MTRN

Execute F159_MTRN to

- specify the amount of data to be sent
- specify the communication port to be used
- output the data from the communication port to the external device.

When the execution condition of F159_MTRN turns to TRUE and the "transmission done" flag is TRUE, transmission starts. (For details on flag operation, see page 121.)

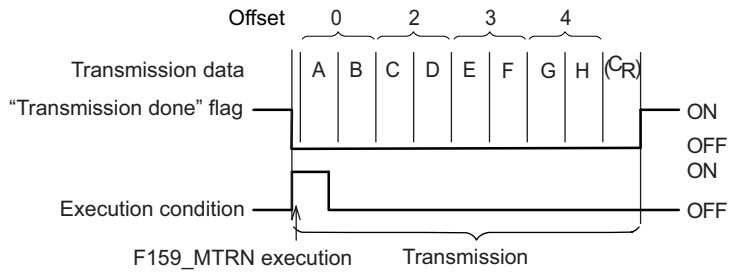
When sending data, operation is as follows:

- The number of bytes to be sent is set in offset 0 of the send buffer.
- The "transmission done" flag turns to FALSE.
- The data in the send buffer is sent starting with the low order byte in offset 1.
- The start and end codes specified in the system registers are automatically added to the data sent.
- During transmission, F159_MTRN cannot be executed again.
- The "reception done" flag turns to FALSE.
- The number of bytes received is set to 0 in offset 0 of the receive buffer.
- Data received is written into the receive buffer

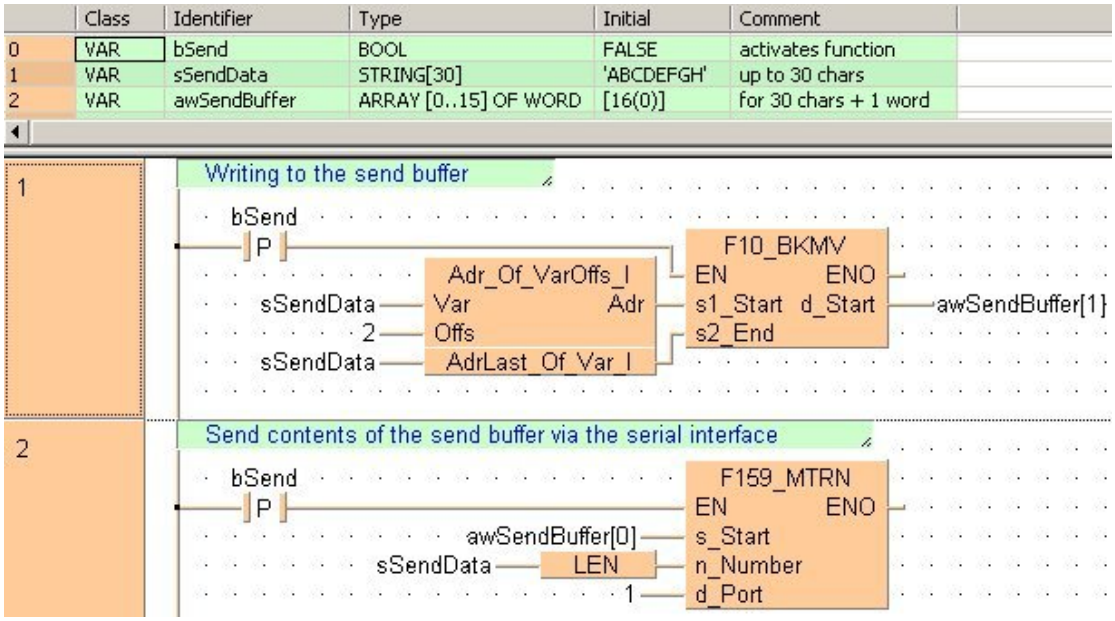
When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE. The end code is automatically added to the data sent. At the end of transmission, the value in offset 0 is automatically reset to 0.

◆EXAMPLE

Transmit the characters "ABCDEFGH" to an external device connected to COM port 1. For start code and end code the default settings "No-STX" and "CR" are selected.



POU Header and LD Body



ST Body

```
if (DF(bSend)) then
    (* Creating the send buffer *)
    F10_BKMV(s1_Start := Adr_Of_VarOffs(Var := sSendData, Offs := 2),
    s2_End := AdrLast_Of_Var(sSendData), d_Start => awSendBuffer[1]);
    (* Send contents of the send buffer via the serial interface *)
    F159_MTRN(s_Start := awSendBuffer[0], n_Number := LEN(sSendData), d_Port := 1);
end_if;
```

When the variable **bSend** is set to TRUE, the function F10_BKMV copies the characters of the string **sSendData** to the buffer **awSendBuffer** beginning at **awSendBuffer[1]**.

Then, F159_MTRN sends the data from the first element of the send buffer (**awSendBuffer[0])** as specified by **s_Start**. The length of the string to be sent (8 bytes) is set at **n_Number** (using the function LEN to calculate the number of bytes). The data is output from COM port 1 as specified by **d_Port.**

◆NOTE

- **For details on the operation of the "reception done" flag, the "transmission done" flag, and the communication error flag, see page 121 .**
- **For details on the format of the data in the send buffer and in the receive buffer, please see "Data Format" on page 120.**
- **Data cannot be sent unless the pin CS (Clear to Send) is on. When connecting to a three-wire port, short-circuit the RS and CS pins.**

7.6.3 Receiving Data from External Devices

Steps for receiving data from external devices:

1. Setting the communication parameters and specifying the receive buffer
2. Receiving the data
3. Processing the data in the receive buffer
4. Preparing the system to receive subsequent data

◆NOTE

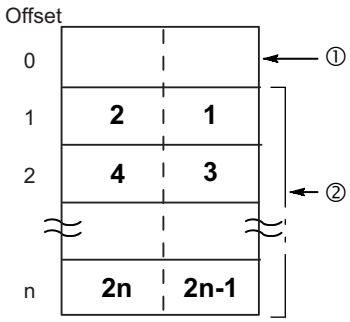
Data received via the communication ports of an MCU has to be moved to the CPU receive buffer using the instruction F161_MRCV.

1. Setting the communication parameters (see "Setting Communication Parameters" on page 111)

2. Receiving the data

Data is automatically received in the receive buffer defined in the system registers. Reception can be controlled by the "reception done" flag or by directly evaluating the receive buffer. (For details on flag operation, see page 121.) When this flag is FALSE and data is sent to the communication port from an external device, operation takes place as follows. (The "reception done" flag turns to FALSE after switching to RUN mode.)

- Incoming data is stored in the receive buffer. Start and end codes are not stored in the receive buffer. The storage area for the data received starts with the second word of the receive buffer (offset 1). Offset 0 contains the number of bytes received. The initial value of offset 0 is 0.



① Storage area for the number of bytes received

② Storage area for the data received

Bold numbers indicate the order of reception.

- When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. The "reception done" flag only turns to TRUE if an end code, e.g. CR, has been selected in the system registers.

3. Processing the data in the receive buffer

- Verify the end of reception.
- Copy the data in the receive buffer to a target area defined in the program using a

transfer instruction, e.g. F10_BKMV.

◆**NOTE**

For details on the operation of the "reception done" flag, see page 121

4. Preparing the system for the reception of further data

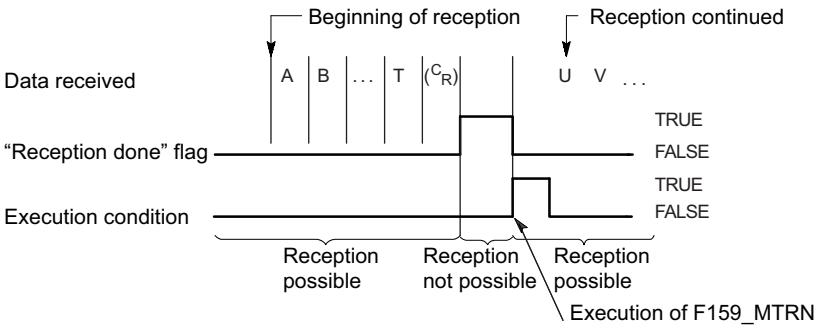
In order to receive the next data, reset the receive buffer. This is done automatically when sending the next data with F159_MTRN:

- Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. Subsequent data will be stored in the receive buffer starting at offset 1. (The receive buffer is not cleared).
- The "reception done" flag turns to FALSE.

To reset the receive buffer without sending further data, execute F159_MTRN with **n_Number** = 0.

◆**EXAMPLE**

Receive a string of 8 bytes containing the characters "ABCDEFGH" via COM port 1. The characters are stored in ASCII HEX code without start and end codes.



Receive buffer layout:

Offset

0	8	
1	16#42(B)	16#41(A)
2	16#44(D)	16#43(C)
3	16#46(F)	16#45(E)
4	16#48(H)	16#47(G)

When reception begins, the value in offset 0 is 8. At the end of reception, the value in offset 0 is 0. The data in offset 1 to offset 4 is received in order from the low order byte.

System register settings:

No	Item Name	Data	Dim...
412	COM port 1 communication mode	Program controlled...	
410	COM port 1 station number	1	
415	COM port 1 baud rate	9600	baud
413	COM port 1 sending data length	8 bits	
413	COM port 1 sending parity check	With-Odd	
413	COM port 1 sending stop bit	1 bit	
413	COM port 1 sending start code	No-STX	
413	COM port 1 sending end code/reception done condition	CR	
416	COM port 1 receive buffer starting address	200	
417	COM port 1 receive buffer capacity	5	
412	COM port 1 modem connection	Disable	

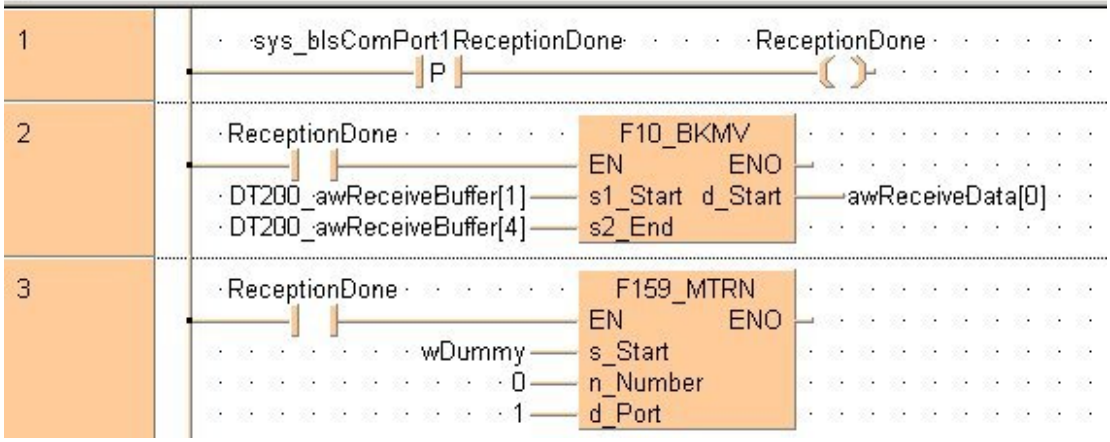
In order to use the data in the receive buffer, define a global variable having the same starting address and capacity. In this example, the starting address is 200 (VAR_GLOBAL ReceivedData) and the receive buffer capacity is 5 (ARRAY [0..4] OF WORD).

GVL

	Class	Identifier	FP A...	IEC Addr...	Type	Initial
0	VAR_GLOBAL	DT200_awReceiveBuffer	DT200	%MW5.200	ARRAY [0..4] OF WORD	[5(0)]

POU Header and LD Body

	Class	Identifier	Type	Initial	Comment
0	VAR	wDummy	WORD	0	
1	VAR	ReceptionDone	BOOL	FALSE	
2	VAR_EXTERNAL	DT200_awReceiveBuffer	ARRAY [0..4] OF WORD	[5(0)]	
3	VAR	awReceiveData	ARRAY[0..3] OF WORD	[4(0)]	



ST Body

```
if (sys_bIsComPort1ReceptionDone) then
    F10_BKMV(s1_Start := DT200_awReceiveBuffer[1], s2_End := DT200_awReceiveBuffer[4],
    d_Start => awReceiveData[0]);
    F159_MTRN(s_Start := wDummy, n_Number := 0, d_Port := 1);
end_if;
```

Data can be received when the "reception done" flag is FALSE. The "reception done" flag is evaluated by the system variable sys_bIsComPort1ReceptionDone. When the reception of the

data is complete (the end code has been received), the "reception done" flag turns to TRUE, and subsequently, receiving data is prohibited. To prepare the system to receive the next data without immediately sending further data, the receive buffer is reset by executing F159_MTRN with **n_Number** = 0.

◆NOTE

- **The status of the "reception done" flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal relay should be copied to a variable at the beginning of the program.**
- **The start code "STX" resets the receive buffer. Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. Subsequent data will be stored in the receive buffer starting at offset 1.**
- **For details on the format of the data in the send buffer and in the receive buffer, please see "Data Format" on page 120.**

7.6.4 Data Format

Remember the following when accessing data in the send and receive buffers:

- The format of the data in the send buffer depends on the data type of the transmission data (e.g. STRING) and on the conversion function used in the PLC program (e.g. F95_ASC). There is no conversion when data in the send buffer is sent.
- The start and end codes specified in the system registers are automatically added to the data sent. The start code is added at the beginning, the end code at the end of the send string. Do not include start or end codes in the send string.
- The format of the data in the receive buffer depends on the data format used by the external device. Use a conversion function to convert the data into the desired format, e.g. F27_AHEX.
- Start and end codes in the data received are recognized if the corresponding start and end codes have been specified in the system registers. Start and end codes are not stored in the receive buffer. The end code serves as a reception done condition, i.e., the "reception done" flag turns to TRUE when the end code is received. The start code resets the receive buffer.
- If "None" is selected for the start code, a start code is not added to the data sent and is not recognized in the data received. Without start code, the receive buffer can only be reset by executing F159_MTRN.
- If "None" is selected for the end code, an end code is not added to the data sent and is not recognized in the data received. Without end code, the "reception done" flag does not turn to TRUE. The end of reception can only be determined by a time-out using the IsReceptionDoneByTimeOut function or by evaluating the data in the receive buffer.

Different end code settings for sending and receiving

Sometimes you do not want to send an end code, but need an end code in the data received to set the "reception done" flag to TRUE. In this case, select the desired end code in the system registers and execute F159_MTRN specifying a negative number for **n_Number**.

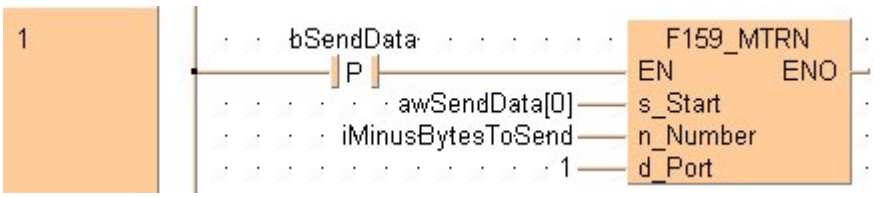
◆EXAMPLE

Send 4 bytes of data without adding an end code:

POU Header

	Class	Identifier	Type	Initial	Comment
0	VAR	bSendData	BOOL	FALSE	
1	VAR_CONST...	iMinusBytesToSend	INT	-6	Negative number: No terminator added!
2	VAR	awSendData	ARRAY [0..3] OF WORD	[4(0)]	First word: Number of bytes sent.
3	VAR				Words 1 to 3: 6 data bytes to send!

LD Body



ST Body

```
if (DF(bSendData)) then
    F159_MTRN(s_Start := awSendData[0], n_Number := iMinusBytesToSend, d_Port

end_if;
```

7.6.5 Flag Operation

Program controlled communication provides for half duplex communication, i.e. communication is possible in both directions, but not simultaneously. Sending can be controlled by the "transmission done" flag. Reception can be controlled by the "reception done" flag or by directly evaluating the receive buffer.

The flags are special internal relays which turn to TRUE or to FALSE under specific conditions. They can be evaluated using special functions or system variables.

"Reception done" flag

When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. F159_MTRN turns the "reception done" flag to FALSE.

The "reception done" flag can be evaluated using the IsReceptionDone function. Or use the system variable sys_bIsComPort1ReceptionDone or sys_bIsToolPortReceptionDone, depending on the port. The end of reception can also be determined by time-out using the IsReceptionDoneByTimeOut function or by checking the contents of the receive buffer.

The status of the "reception done" flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal relay should be copied to a variable at the beginning of the program.

Port name	Port number	Special internal relay	Function name	System variable name	Bit status
TOOL	0	R903E	IsReceptionDone	sys_bIsToolPortReceptionDone	TRUE
COM1	1	R9038		sys_bIsComPort1ReceptionDone	

"Transmission done" flag

When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE. New data may be sent or received. F159_MTRN turns the "transmission done" flag to FALSE. While F159_MTRN is executed, no data can be received.

The "transmission done" flag can be evaluated using the IsTransmissionDone function. Or use the system variable sys_bIsComPort1TransmissionDone or sys_bIsToolPortTransmissionDone, depending on the port.

Port name	Port number	Special internal relay	Function name	System variable name	Bit status
TOOL	0	R903F	IsTransmissionDone	sys_bIsToolPortTransmissionDone	TRUE
COM1	1	R9039		sys_bIsComPort1TransmissionDone	

Communication error flag

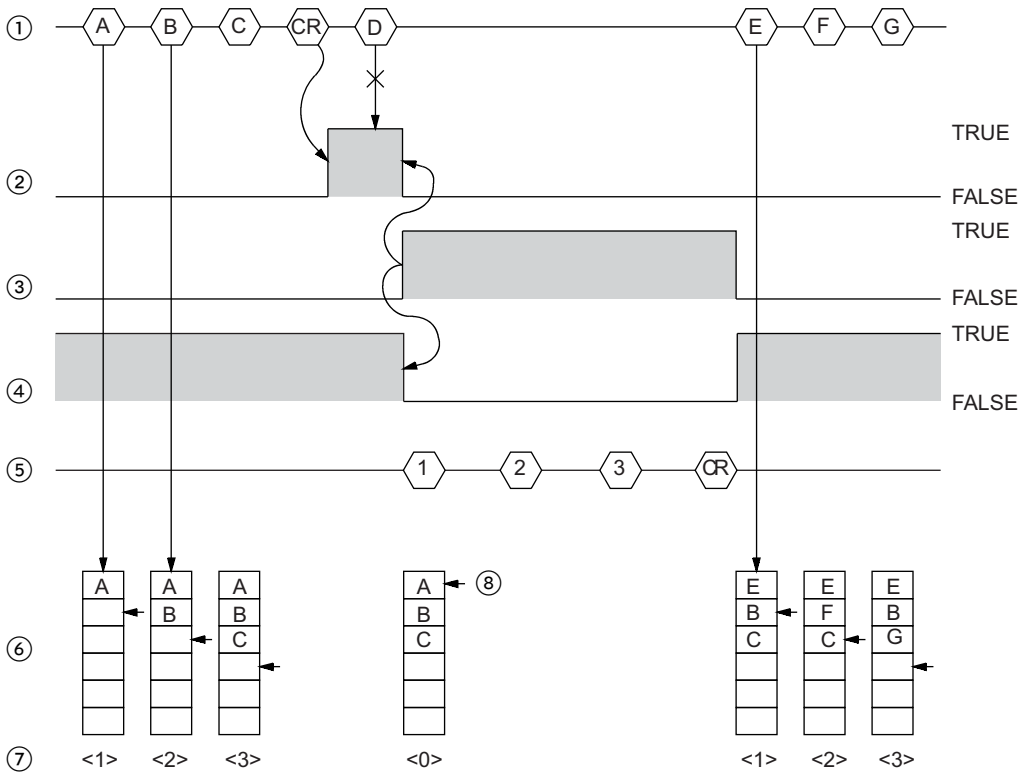
If the communication error flag turns to TRUE during reception, reception continues. Execute F159_MTRN to turn the error flag to FALSE and to move the write pointer back to offset 1.

The communication error flag can be evaluated using the IsCommunicationError function. Or use the system variable sys_bIsComPort1CommunicationError or sys_bIsToolPortCommunicationError, depending on the port.

Port name	Port number	Special internal relay	Function name	System variable name	Bit status
TOOL	0	R900E	IsCommunicationError	sys_bIsToolPortCommunicationError	TRUE
COM1	1	R9037		sys_bIsComPort1CommunicationError	

7.6.5.1 Start Code: No-STX; End Code: CR

Receiving and sending data:



①	Data received from external device	⑤	Data sent to external device
②	"Reception done" flag	⑥	Receive buffer
③	F159_MTRN execution	⑦	Number of bytes received
④	"Transmission done" flag	⑧	Write pointer

When receiving data, operation is as follows:

1. Characters A, B, and C received from the external device are stored in the receive buffer.
2. When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. (Character D is not stored.)
3. F159_MTRN is executed to send response data to the external device. When F159_MTRN is executed:
   - The receive buffer is reset.
   - The "reception done" flag turns to FALSE.
   - The "transmission done" flag turns to FALSE.
   - The communication error flag turns to FALSE.
   - Characters 1, 2, and 3 are sent to the external device.

- The end code is automatically added to the data sent.
- While F159_MTRN is being executed, no data can be received. (The "transmission done" flag is FALSE.)

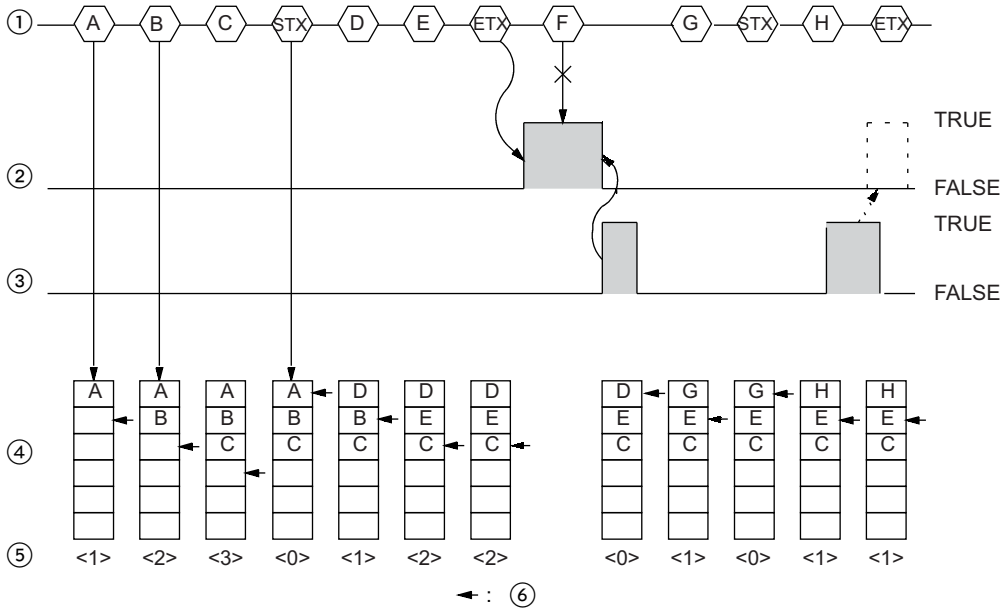
4. When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE.
5. Characters E, F, and G received from the external device are stored in the receive buffer.

◆NOTE

Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. Subsequent data will be stored in the receive buffer starting at offset 1.

7.6.5.2 Start Code: STX; End Code: ETX

Receiving data:



①	Data received from external device	④	Receive buffer
②	"Reception done" flag	⑤	Number of bytes received
③	F159_MTRN execution	⑥	Write pointer

When receiving data, operation is as follows:

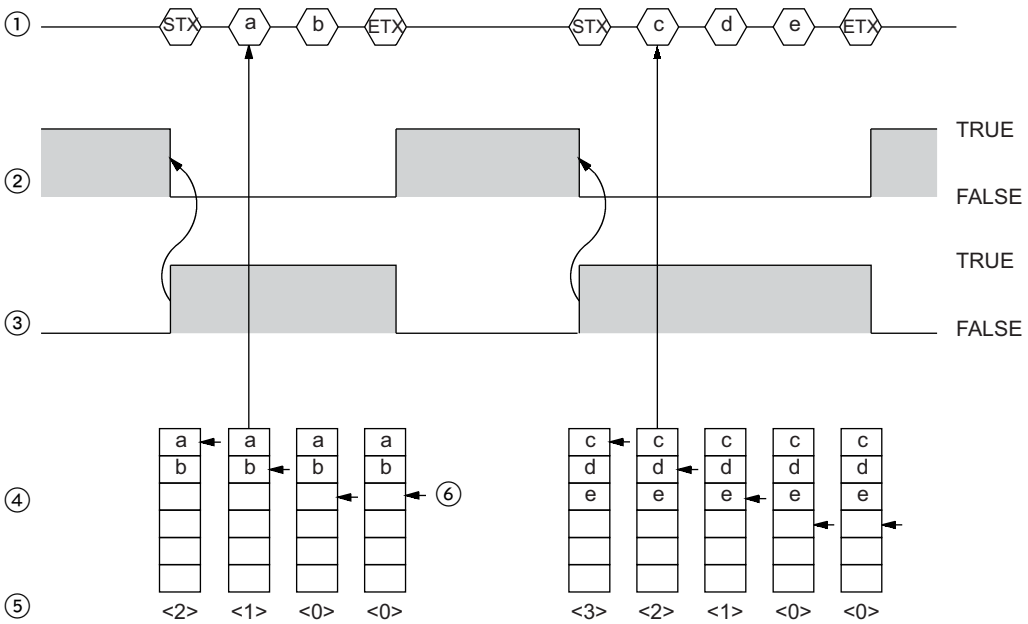
1. Characters A, B, and C received from the external device are stored in the receive buffer.
2. The start code "STX" resets the receive buffer.

3. Characters D and E received from the external device are stored in the receive buffer.
4. When the end code is received, the "reception done" flag turns to TRUE. Reception of any further data is prohibited. (Character F is not stored.)
5. When F159_MTRN is executed:
   - The number of bytes received is set to 0 in offset 0 of the receive buffer.
   - The "reception done" flag turns to FALSE.
   - The number of bytes received is set to 0 in offset 0 of the receive buffer. (Character G is stored.)
6. The start code "STX" resets the receive buffer. (Character H is stored.)
7. F159_MTRN is executed at the same time the end code is received from the external device. F159_MTRN turns the "reception done" flag to FALSE. Therefore, this flag will not be detected.

◆NOTE

- **Resetting the receive buffer sets the number of bytes received in offset 0 to 0 and moves the write pointer back to offset 1. Subsequent data will be stored in the receive buffer starting at offset 1.**
- **If two start codes are received from the external device, data following the second start code overwrites the data in the receive buffer.**

Sending data:



①	Data to be sent	④	Send buffer
②	"Transmission done" flag	⑤	Number of bytes to be sent
③	F159_MTRN execution	⑥	Write pointer

When sending data, operation is as follows:

F159_MTRN is executed to send data to the external device. When F159_MTRN is executed:

1. The "transmission done" flag turns to FALSE.
2. The start code is sent automatically.
3. The number of bytes to be sent is set in offset 0 of the send buffer.
4. The characters a and b are sent to the external device.
   - The end code is automatically added to the data sent.
   - While F159_MTRN is being executed, no data can be received. (The "transmission done" flag is FALSE.)
5. When the specified number of bytes has been sent, the "transmission done" flag turns to TRUE.
6. Now, F159_MTRN can be executed again. When F159_MTRN is executed:
   - Steps 1 to 5 are repeated. This time, the characters c, d, and e are sent.

7.6.6 1:1 Communication

System register settings

By default, the COM port is set to MEWTOCOL-COM mode. For 1:1 program controlled communication, the system registers should be set as shown below.

- Settings for COM port 1 (or TOOL port)

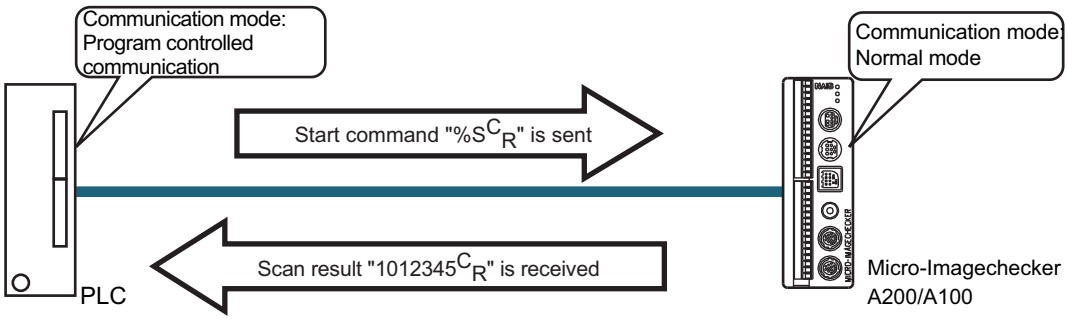
No.	Name	Set value
412	COM port 1 - communication mode	Program controlled
413	COM port 1 - communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
415	COM port 1 - baud rate	2400–115200bit/s
416 (420)	COM port 1 - receive buffer starting address	0–32764 (factory setting: 0) (see note)
417 (421)	COM port 1 - receive buffer capacity	0–2048 words (factory setting: 2048 words)

◆**NOTE**

When using C10, C14, or C16, the range is 0–12312.

7.6.6.1 1:1 Communication with Micro-Imagechecker

The FP0R and Micro-Imagechecker A100/A200 are connected using an RS232C cable. The results of the scan are stored in the data registers of the FP0R.



1:1 communication between the FP0R and a Micro-Imagechecker

After the scan start code "%S^{C}_{R}" has been sent from the FP0R side, the scan result is returned from the Micro-Imagechecker as the response.

Communication format settings for Micro-Imagechecker A100/A200

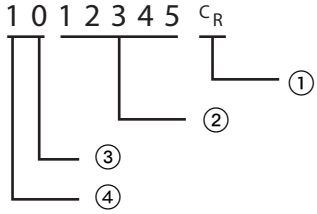
To set the communication mode and transmission format settings for the Micro-Imagechecker, select "5: Communication" under "5: ENVIRONMENT" on the main menu, and set the following items.

No.	Name	Set value	
51	Communication mode	Normal Mode	
52	RS232C	Baud rate: Length: Stop bit: Parity: Flow Control:	9600bit/s 8 1 Odd None
53	Serial Output	Output: Invalid Digit: Read End: Process End: Numerical Calculation: Judgment:	5 Column Repl. 0 None None Output Output

◆NOTE

- **If "Del" is specified for the invalid processing parameter, zero suppression processing will be carried out on the output data, and the output format will be changed. Always make sure "Repl. 0" is specified.**
- **When outputting data to an external device, numerical calculation is required, so "Out" should be specified for the "Numerical calculation" parameter.**

With the above settings, the following data will be output from the Micro-Imagechecker:



①	Terminator (end code)	③	Judgment output No. 2 0=NG
②	Results of numerical calculation No. 1	④	Judgment output No. 1 1=OK

A100/A200 COM port

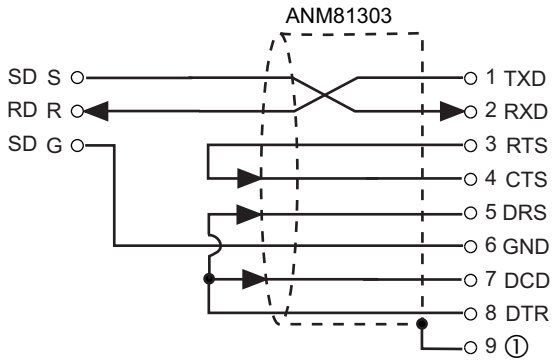
Please make the connection using a special RS232C cable available for the Micro-Imagechecker (order no. ANM81303).

A100/A200 COM port

Pin	Name	Cable color
1	TXD	Red
2	RXD	White
3	RTS	Black
4	CTS	Yellow
5	DSR	Blue
6	GND	Green
7	CD	Brown
8	DTR	Gray

Wiring diagrams

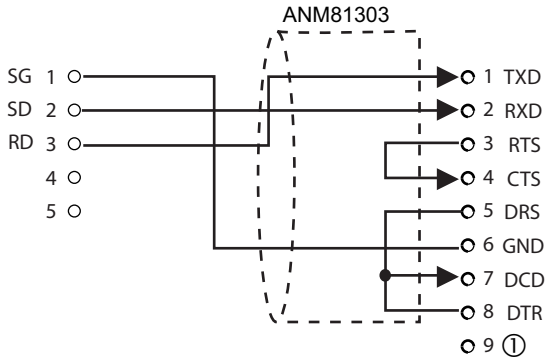
- Using the COM port



① Shield (housing)

Left: FP0R, right: Micro-Imagechecker

- Using the TOOL port

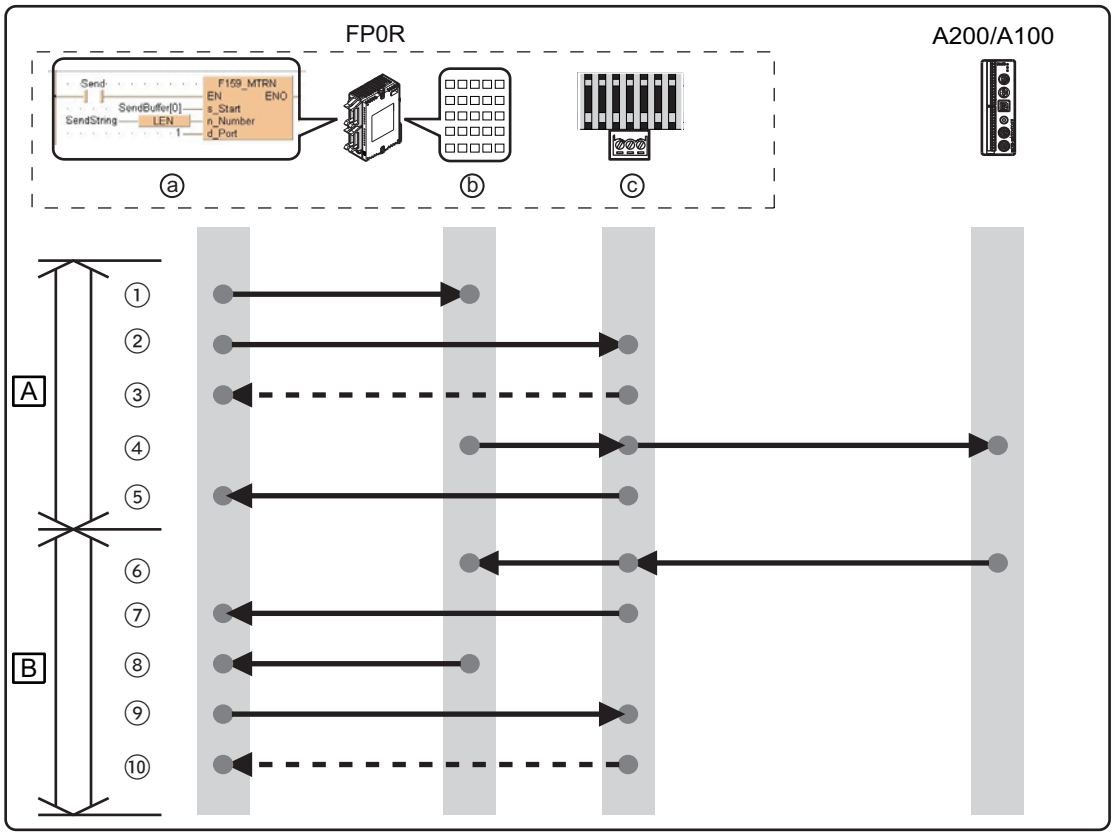


① Shield (housing)

Left: FP0R, right: Micro-Imagechecker

Procedure of communication

In the following example, the Micro-Imagechecker is connected to COM port 1.



ⓐ FPWIN Pro program ⓑ Data registers ⓒ RS232C port

A Sending data	①	Write start command "%S^{C}_{R}" in send buffer
	②	Send data with F159_MTRN
	③	"Transmission done" flag: FALSE, "Reception done" flag: FALSE, Receive buffer reset
	④	Send start command "%S^{C}_{R}"
	⑤	"Transmission done" flag: TRUE
B Receiving data	⑥	Receive scan result "1012345$^{C}_{R}$"
	⑦	"Reception done" flag: TRUE
	⑧	Read data "1012345$^{C}_{R}$"
	⑨	Send empty data with F159_MTRN
	⑩	"Transmission done" flag: FALSE, "Reception done" flag: FALSE, Receive buffer reset

System register settings:

No	Item Name	Data	Dim...
412	COM port 1 communication mode	Program controlled...	
410	COM port 1 station number	1	
415	COM port 1 baud rate	9600	baud
413	COM port 1 sending data length	8 bits	
413	COM port 1 sending parity check	With-Odd	
413	COM port 1 sending stop bit	1 bit	
413	COM port 1 sending start code	No-STX	
413	COM port 1 sending end code/reception done condition	CR	
416	COM port 1 receive buffer starting address	200	
417	COM port 1 receive buffer capacity	5	
412	COM port 1 modem connection	Disable	

In order to use the data in the receive buffer, define a global variable having the same starting address and capacity. In this example, the starting address is 200 (VAR_GLOBAL awReceiveBuffer) and the receive buffer capacity is 5 (ARRAY [0..4] OF WORD).

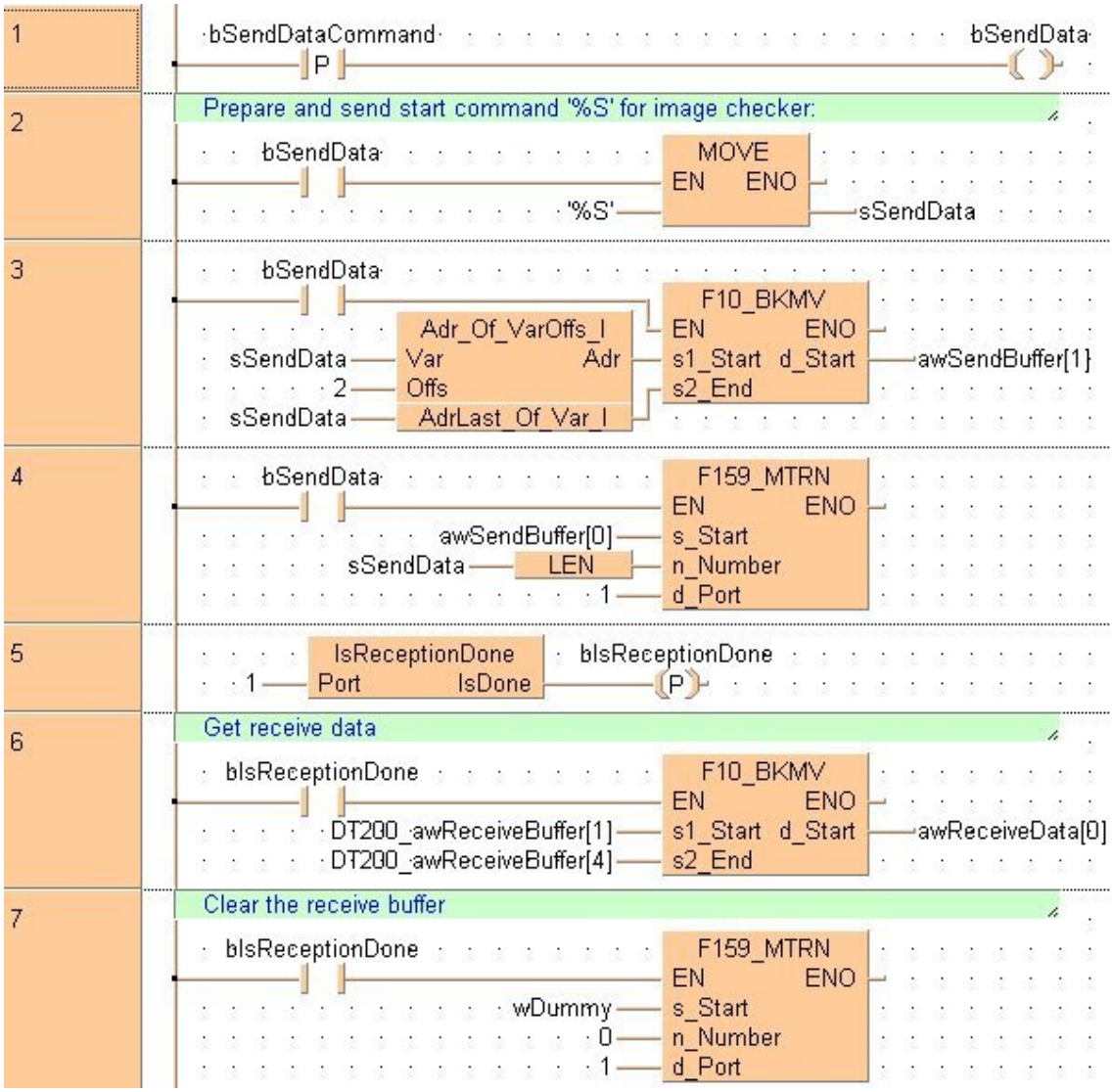
GVL

	Class	Identifier	FP A...	IEC Addr...	Type	Initial
0	VAR_GLOBAL	DT200_awReceiveBuffer	DT200	%MW5.200	ARRAY [0..4] OF WORD	[5(0)]

POU Header

	Class	Identifier	Type	Initial
0	VAR	bSendDataCommand	BOOL	FALSE
1	VAR	bSendData	BOOL	FALSE
2	VAR	sSendData	STRING[2]	''
3	VAR	awSendBuffer	ARRAY [0..1] OF WORD	[2(0)]
4	VAR	bIsReceptionDone	BOOL	FALSE
5	VAR	awReceiveData	ARRAY [0..3] OF WORD	[4(0)]
6	VAR	wDummy	WORD	0
7	VAR_EXTERNAL	DT200_awReceiveBuffer	ARRAY [0..4] OF WORD	[5(0)]

LD Body



ST Body

```
if (DF(bSendDataCommand)) then
    (* Prepare and send start command '%S' for image checker: *)
    sSendData:='%S';
    F10_BKMV(s1_Start:=Adr_Of_VarOffs(Var:=sSendData, Offs:=2), s2_End:=AdrLast_Of_Var(sSendData),
    d_Start=>awSendBuffer[1]);
    F159_MTRN(s_Start := awSendBuffer[0], n_Number := LEN(sSendData), d_Port := 1);
end_if;

if (DF(IsReceptionDone(Port:=1))) then
    (* Get receive data *)
    F10_BKMV(s1_Start := DT200_awReceiveBuffer[1], s2_End := DT200_awReceiveBuffer[4],
    d_Start => awReceiveData[0]);
    (* Clear the receive buffer *)
    F159_MTRN(s_Start := wDummy, n_Number := 0, d_Port := 1);
end_if;
```

The status of the "reception done" flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal relay should be copied to a variable at the beginning of the program.

Buffer statuses

The following shows the statuses of the send and receive buffers when the sample program is run.

Status of the send buffer before sending:

Offset	2	
0		
1	16#53(S)	16#25(%)

At the end of transmission, the value in offset 0 is automatically reset to 0.

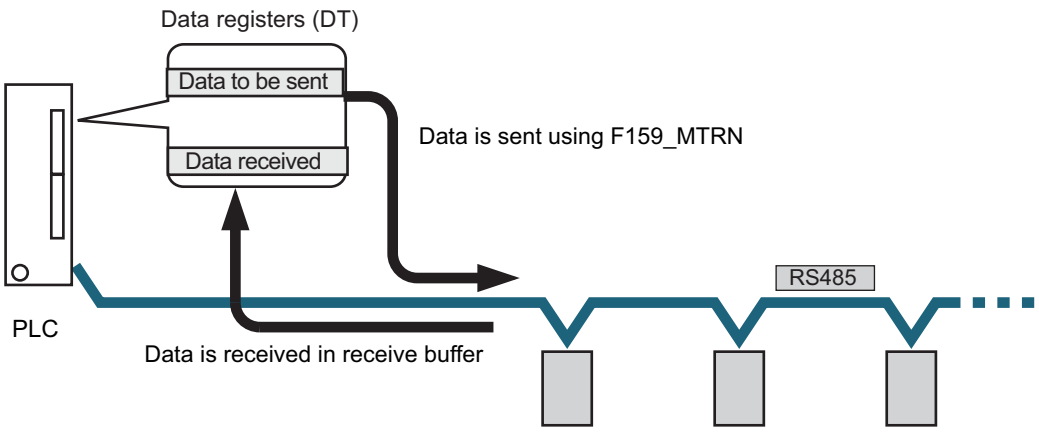
Status of the receive buffer when the reception is completed:

	7	
DT200		
DT201	16#30 (0)	16#31 (1)
DT202	16#32 (2)	16#31 (1)
DT203	16#34 (4)	16#33 (3)
DT204		16#35 (5)

The number of bytes received is stored in offset 0. The received data is stored in order from the low-order byte.

7.6.7 1:N Communication

The FP0R and the external units are connected using an RS485 cable. Using the protocol that matches the external units, the F159_MTRN instruction is used to send and receive data.



1:N communication between the FP0R and several external devices

System register settings

By default, the COM port is set to MEWTOCOL-COM mode. For 1:N program controlled communication, the system registers should be set as shown below.

- Settings for COM port 1 (or TOOL port)

No.	Name	Set value
412	COM port 1 - communication mode	Program controlled
413	COM port 1 - communication format [1)]	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
415	COM port 1 - baud rate [1)]	2400–115200bit/s
416 (420)	COM port 1 - receive buffer starting address	0–32762 (factory setting: 0) (see note)
417 (421)	COM port 1 - COM port 2	0–2048 words (factory setting: 2048 words)

[1)] The setting must match the external device connected to the communication port.

7.6.8 Setting Communication Parameters

Make sure that the PLC type selected in FPWIN Pro is "FP0".

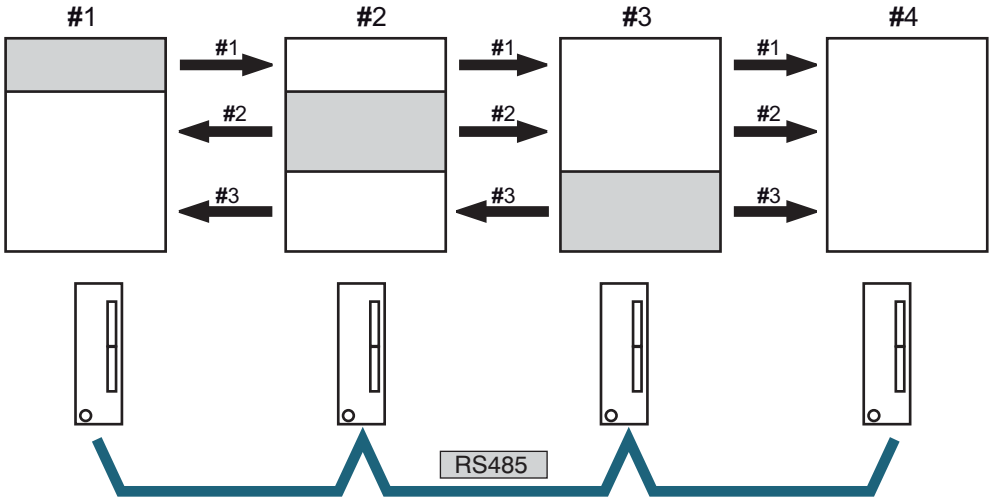
In the FP0 compatibility mode, the instruction F144_TRNS is used instead of F159_MTRN.

◆REFERENCE

For details on the instruction F144_TRNS, please refer to the online help of FPWIN Pro.

7.7 PLC Link

PLC Link is an economic way of linking PLCs using a twisted-pair cable and the MEWNET protocol. Data is shared with all PLCs by means of dedicated internal relays called link relays (L) and data registers called link registers (LD). The statuses of the link relays and link registers of one PLC are automatically fed back to the other PLCs on the same network. The link relays and link registers of the PLCs contain areas for sending and areas for receiving data. Station numbers and link areas are allocated using the system registers.

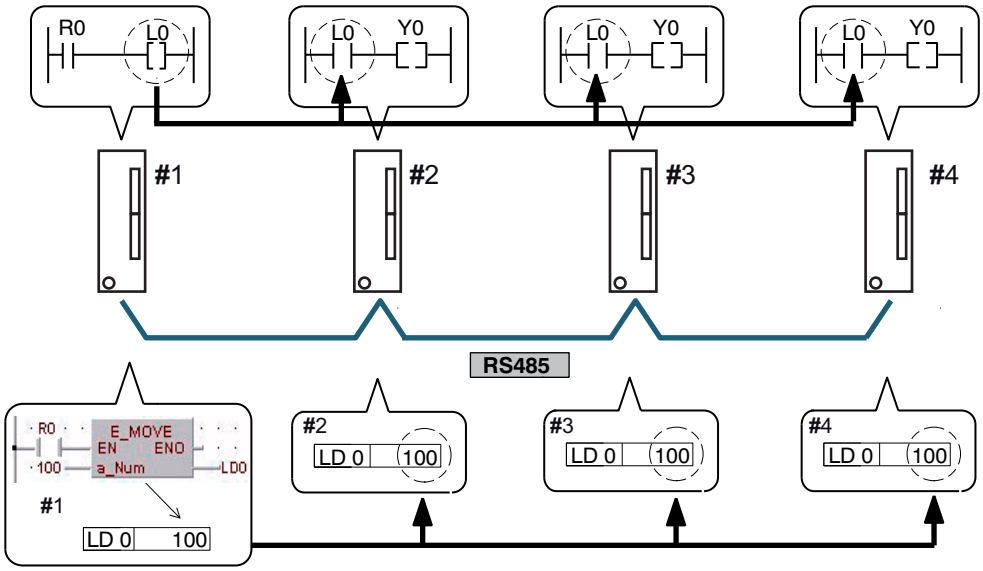


Sharing of data in a PLC link using dedicated send and receive areas

	Send area		Receive area	#	Station number of PLC

◆EXAMPLE

Link relay L0 for station #1 turns to TRUE. The status change is fed back to the programs of the other stations, and Y0 of the other stations is set to TRUE. A constant of 100 is written to link register LD0 of station #1. The contents of LD0 in the other stations are also changed to a constant of 100.



PLC Link connection between four FP0R units

#	Station number of PLC	LD	Link register

7.7.1 Setting Communication Parameters

Make the following settings for the communication port:

- communication mode (PLC Link)
- station number
- link area

For details on setting the communication parameters, see "Setting System Registers in PROG Mode" on page 94. For details on setting the link area, see "Link Area Allocation" on page 138.

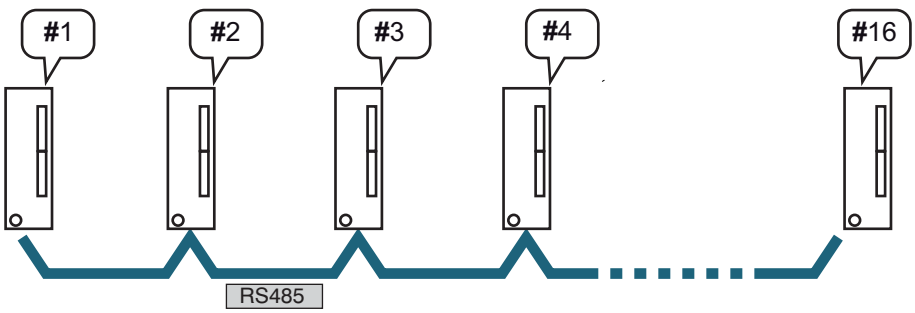
◆NOTE

- **PLC Link is only available for COM port 1.**
- **For RS232C connections, the maximum number of stations is 2.**
- **For PLC Link, the communication format and baud rate settings are fixed:**

Data length:	8 bits
Parity:	Odd
Stop bit:	1 bit
End code:	CR
Start code:	No STX
Baud rate:	115200bit/s

Station number setting for a PLC link

The station number can be set within a range of 1 to 16. For details on setting station numbers,.



Station number of PLC

A maximum of 16 stations can be connected in a PLC link

◆NOTE

- **Make sure the same station number is not used for more than one of the PLCs connected through the PLC Link function.**

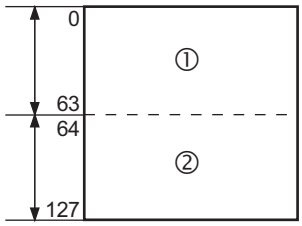
- **Station numbers should be set sequentially and consecutively, starting from 1, with no breaks between them. If there are fewer than 16 stations linked, set the highest station number to reduce the link transmission cycle time. See "Setting the Highest Station Number for a PLC Link" on page 144.**

7.7.2 Link Area Allocation

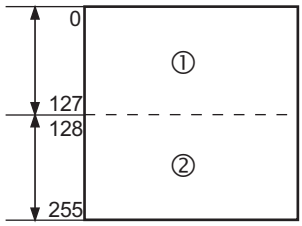
To use the PLC link function, link areas need to be allocated. Set the allocations for both the link relays and link registers using the system registers of the CPU.

Link areas consist of link relays and link registers and are divided into areas for PLC link 0 and PLC link 1. A maximum of 1024 link relays (points) and 128 link registers (words) can be used in the PLC link areas.

Link relays



Link registers



Unit: words

①	For PLC link 0: 1024 points (1st half)	①	For PLC link 0: 128 words (1st half)
②	For PLC link 1: 1024 points (2nd half)	②	For PLC link 1: 128 words (2nd half)

System registers

No.		Name	Default value	Set values
	46	PLC Link 0 and 1 allocation setting	Normal	Normal: 1st half Reverse: 2nd half
PLC link 0	40	Link relays - Send/receive area - Number of words shared by all linked PLCs	0	0-64 words
	41	Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0-128 words
	42	Link relays - Send area - Start sending from this word address	0	0-63
	43	Link relays - Send area - Number of words to send	0	0-64 words
	44	Link registers - Send area - Start sending from this word address	0	0-127
	45	Link registers - Send area - Number of words to send	0	0-128 words
	47[1)]	Highest station number in network	16	1-16
PLC link 1	50	Link relays - Send/receive area - Number of words shared by all linked PLCs	0	0-64 words
	51	Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0-128 words
	52	Link relays - Send area - Start sending from this word	64	64-127

No.		Name	Default value	Set values
		address		
	53	Link relays - Send area - Number of words to send	0	0-64 words
	54	Link registers - Send area - Start sending from this word address	128	128-255
	55	Link registers - Send area - Number of words to send	0	0-128 words
	57[1)]	Highest station number in network	0	0-16

[1)] Set the same value for all PLCs in the link.

◆NOTE

Use the SYS2 instruction to set the link area in RUN mode. Please refer to the FPWIN Pro online help for detailed information.

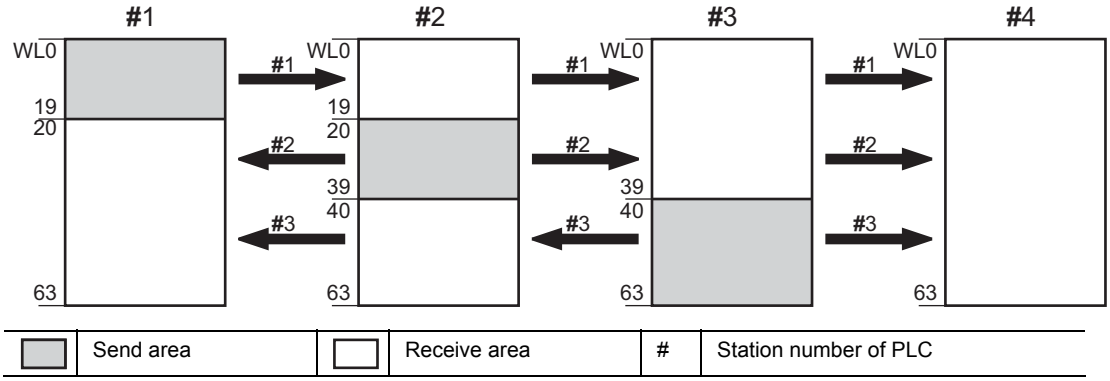
Using PLC link 1

You can either use PLC link 0 or PLC link 1. Set system register 46 to "Reverse" to use PLC link 1. See "PLC Link 0 and 1 Allocation Setting" on page 144.

7.7.2.1 Example for PLC link 0

The PLC link areas are divided into send and receive areas. The link relays and link registers are transmitted from the send area to the receive area of a different PLC. The link relays and registers in the receive area on the receiving side must be within the same area as on the sending side.

Link relay allocation

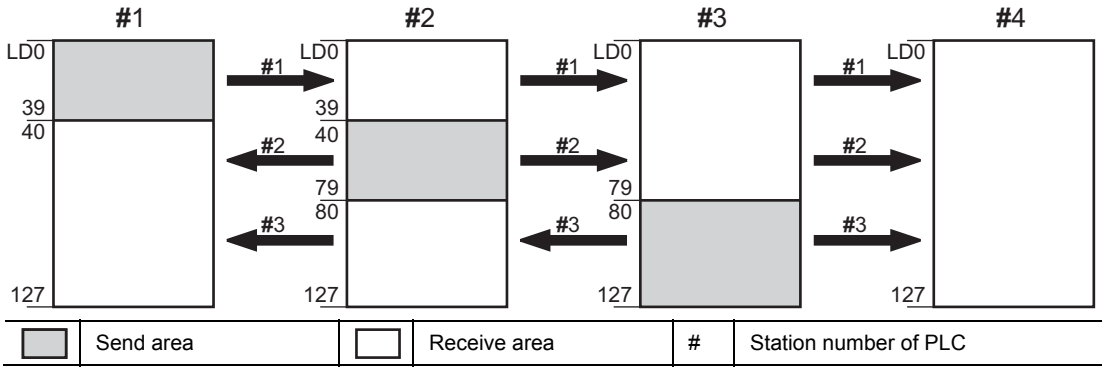


System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
40[1)]	Link relays - Send/receive area - Number of words shared by all linked PLCs	64	64	64	64
42	Link relays - Send area - Start sending from this word address	0	20	40	0
43	Link relays - Send area - Number of words to send	20	20	24	0

1) The value of this system register must be identical for all stations.

Link register allocation



System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
41[1)]	Link registers - Send/receive area - Number of words shared by all linked PLCs	128	128	128	128
44	Link registers - Send area - Start sending from this word address	0	40	80	0
45	Link registers - Send area - Number of words to send	40	40	48	0

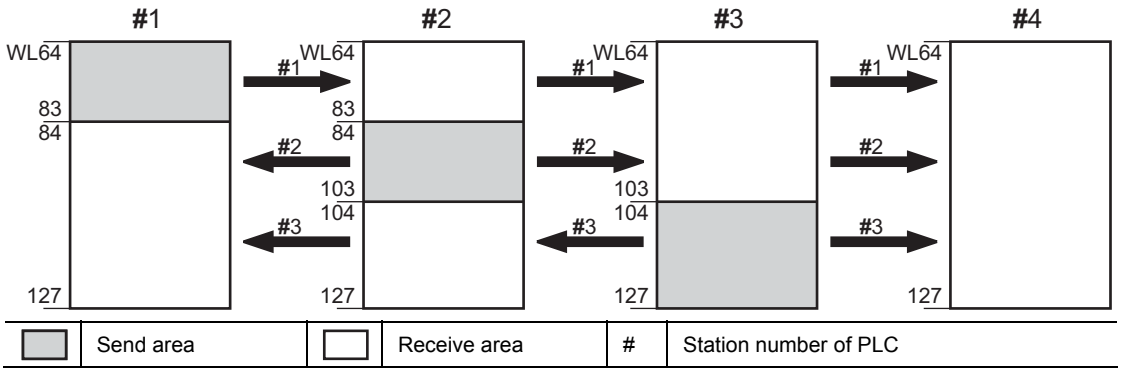
1) The value of this system register must be identical for all stations.

When link areas are allocated as shown above, the send area of station no. 1 can be transmitted to the receive areas of stations no. 2, 3, and 4. Also, the receive area of station no. 1 can receive data from the send areas of stations no. 2 and 3. Station no. 4 is allocated as a receive area only and can receive data from stations no. 1, 2, and 3, but cannot send data to other stations.

7.7.2.2 Example for PLC link 1

Set system register 46 to "Reverse" to use PLC link 1. See "PLC Link 0 and 1 Allocation Setting" on page 144.

Link relay allocation

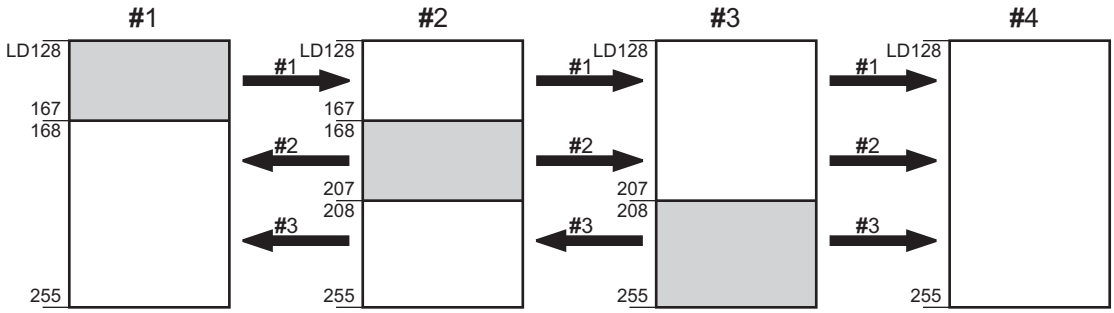


System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
50[1)]	Link relays - Send/receive area - Number of words shared by all linked PLCs	64	64	64	64
52	Link relays - Send area - Start sending from this word address	64	84	104	64
53	Link relays - Send area - Number of words to send	20	20	24	0

1) The value of this system register must be identical for all stations.

Link register allocation



System register settings

No.	Name	Station settings			
		#1	#2	#3	#4
51[1)]	Link registers - Send/receive area - Number of words shared by all linked PLCs	128	128	128	128
54	Link registers - Send area - Start sending from this word address	128	168	208	128
55	Link registers - Send area - Number of words to send	40	40	48	0

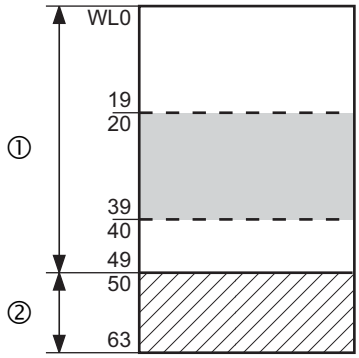
1) The value of this system register must be identical for all stations.

When link areas are allocated as shown above, the send area of station no. 1 can be transmitted to the receive areas of stations no. 2, 3, and 4. Also, the receive area of station no. 1 can receive data from the send areas of stations no. 2 and 3. Station no. 4 is allocated as a receive area only and can receive data from stations no. 1, 2, and 3, but cannot send data to other stations.

7.7.2.3 Partial Use of Link Areas

In the link areas available for PLC link, link relays with a total of 1024 points (64 words) and link registers with a total of 128 words can be used. This does not mean, however, that it is necessary to reserve the entire area. Parts of the area which have not been reserved can be used as internal relays and internal registers.

Link relay allocation



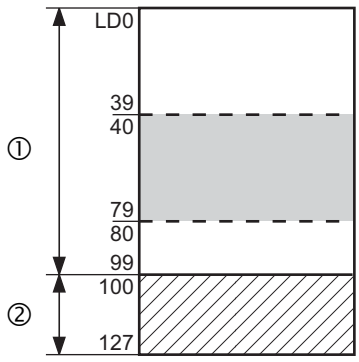
	Send area
	Receive area
	Area for internal relays
①	Used for link relays
②	Not used for link relays

System register settings

No.	Name	#1
40	Link relays - Send/receive area - Number of words shared by all linked PLCs	50
42	Link relays - Send area - Start sending from this word address	20
43	Link relays - Send area - Number of words to send	20

With the above settings for station number 1, the 14 words (224 points) consisting of WL50 to WL63 can be used as internal relays.

Link register allocation



	Send area
	Receive area
	Area for internal relays
①	Used for link relays
②	Not used for link relays

System register settings

No.	Name	#1
41	Link registers - Send/receive area - Number of words shared by all linked PLCs	100
44	Link registers - Send area - Start sending from this word address	40
45	Link registers - Send area - Number of words to send	40

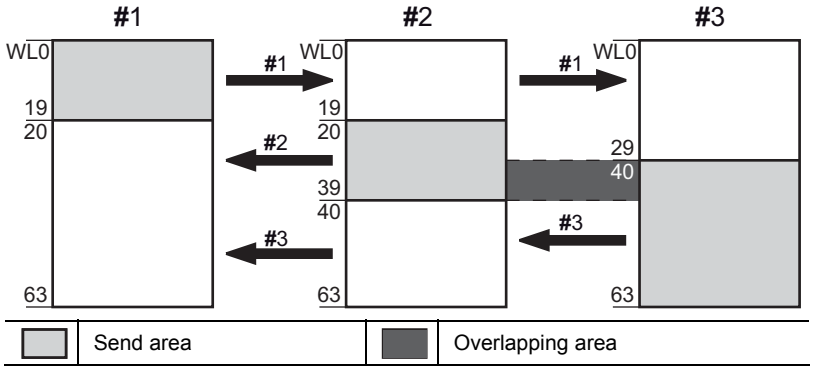
With the above settings for station number 1, the 28 words consisting of LD100 to LD127 can be used as internal registers.

7.7.2.4 Precautions for Allocating Link Areas

A mistake in the link area allocation will cause an error, and communication will be disabled.

Avoid overlapping send areas

When sending data from the send area to the receive area of another PLC, send and receive areas must match. In the example shown below, there is an overlapping area between units no. 2 and 3, and this will cause an error, so that communication cannot be carried out.



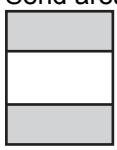
System register settings

No.	Name	Station settings		
		#1	#2	#3
40	Link relays - Send/receive area - Number of words shared by all linked PLCs	64	64	64
42	Link relays - Send area - Start sending from this word address	0	20	30
43	Link relays - Send area - Number of words to send	20	20	34

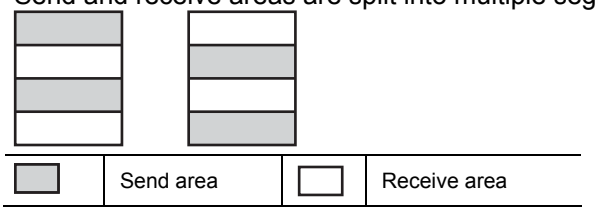
Invalid allocations

The following allocations are not possible, neither for link relays nor for link registers:

- Send area is split



- Send and receive areas are split into multiple segments



7.7.3 Setting the Highest Station Number for a PLC Link

Station numbers should be set sequentially and consecutively, starting from 1, with no breaks between them. If there is a missing station number, or if there is a station for which the power supply has not been turned on, the response time for the PLC link (the link transmission cycle time) will be longer (see page 146).

If there are fewer than 16 stations linked, set the highest station number to reduce the link transmission cycle time. (The default value is 16.) Set the same value for all PLCs in the link.

The highest station number is set using system register no. 47 for PLC link 0 or system register no. 57 for PLC link 1.

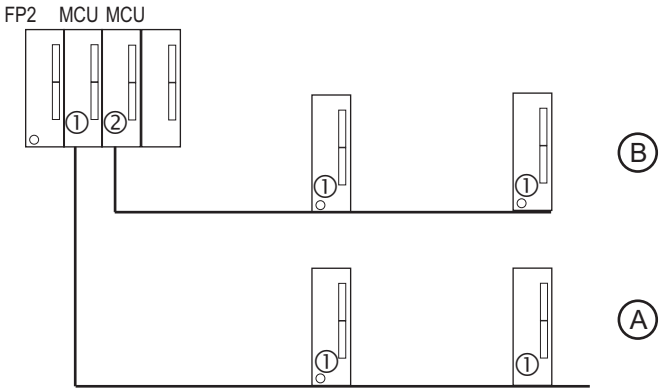
Sample settings

Total number of stations	2		4				n
Station number	1	2	1	2	3	4	n
Highest station number [1)]	2	2	4	4	4	4	N

[1)] Same setting for each station

7.7.4 PLC Link 0 and 1 Allocation Setting

The default setting of system register 46 (PLC Link 0 and 1 allocation setting) is "Normal". This means that the unit which is closest to the CPU uses PLC link 0 and the unit which is further away uses PLC link 1. To reverse this behavior, select "Reverse". In the example below, selecting "Reverse" for the PLCs in PLC link 1 (B) has the advantage that the user does not have to make any address conversions between these PLCs and the MCU. Instead, the same numbers may be used for all PLCs in the link.



① In the default setting ("Normal"), the first half of the link relays and link registers is used (WL0-WL63, LD0-LD127).

② In the default setting ("Normal"), the second half of the link relays and link registers is used (WL64-WL127, LD 128-LD225).

Ⓐ PLC link 0

Ⓑ PLC link 1 - Set system register 46 to "Reverse" in the PLCs of PLC link 1.

7.7.5 Monitoring

When using a PLC link, the operation status of the links can be monitored using the relays below. In FPWIN Pro, choose **Monitor** → **Special Relays and Registers** → **PLC Link Status** to view the status of each relay.

To monitor other PLC link status items, such as the transmission cycle time and the number of times that errors have occurred, choose **Monitor** → **PLC Link Status** in FPWIN Pro.

Remote programming of other linked PLCs is not possible.

◆NOTE

To access special data registers and special internal relays, use the PLC-independent system variables.

Transmission assurance relays

- For PLC link 0: R9060 to R906F (correspond to station no. 1 to 16)
- For PLC link 1: R9080 to R908F (correspond to station no. 1 to 16)

Before using the data from a different station in the network, check to make sure the transmission assurance relay for this station is TRUE.

Relay no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9060	1	sys_bIsPlcLink0Station1Active	TRUE: • if the PLC link is normal FALSE: • if transmission has been stopped, or • if a problem has occurred, or • if a PLC link is not being used
R9061	2	sys_bIsPlcLink0Station2Active	
R9062	3	sys_bIsPlcLink0Station3Active	
R9063	4	sys_bIsPlcLink0Station4Active	
R9064	5	sys_bIsPlcLink0Station5Active	
R9065	6	sys_bIsPlcLink0Station6Active	
R9066	7	sys_bIsPlcLink0Station7Active	
R9067	8	sys_bIsPlcLink0Station8Active	
R9068	9	sys_bIsPlcLink0Station9Active	
R9069	10	sys_bIsPlcLink0Station10Active	
R906A	11	sys_bIsPlcLink0Statio11Active	
R906B	12	sys_bIsPlcLink0Station12Active	
R906C	13	sys_bIsPlcLink0Station13Active	
R906D	14	sys_bIsPlcLink0Station14Active	
R906E	15	sys_bIsPlcLink0Station15Active	
R906F	16	sys_bIsPlcLink0Station16Active	

Operation mode relays

- For PLC link 0: R9070 to R907F (correspond to station nos. 1 to 16)
- For PLC link 1: R9090 to R909F (correspond to station nos. 1 to 16)

The operation modes (RUN/PROG) can be checked for any given PLC.

Relay no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9070	1	sys_bIsPlcLink0Station1InRunMode	TRUE: • if the unit is in RUN mode FALSE: • if the unit is in PROG mode
R9071	2	sys_bIsPlcLink0Station2InRunMode	
R9072	3	sys_bIsPlcLink0Station3InRunMode	
R9073	4	sys_bIsPlcLink0Station4InRunMode	
R9074	5	sys_bIsPlcLink0Station5InRunMode	
R9075	6	sys_bIsPlcLink0Station6InRunMode	
R9076	7	sys_bIsPlcLink0Station7InRunMode	
R9077	8	sys_bIsPlcLink0Station8InRunMode	
R9078	9	sys_bIsPlcLink0Station9InRunMode	
R9079	10	sys_bIsPlcLink0Station10InRunMode	
R907A	11	sys_bIsPlcLink0Station11InRunMode	
R907B	12	sys_bIsPlcLink0Station12InRunMode	
R907C	13	sys_bIsPlcLink0Station13InRunMode	
R907D	14	sys_bIsPlcLink0Station14InRunMode	
R907E	15	sys_bIsPlcLink0Station15InRunMode	
R907F	16	sys_bIsPlcLink0Station16InRunMode	

PLC link transmission error relay R9050

This relay turns to TRUE if a problem is detected during transmission.

Relay no.	Station no.	System variable name	Conditions for TRUE/FALSE
R9050	1–16	sys_bIsPlcLink0TransmissionError	TRUE: • if a transmission error has occurred in the PLC link, or • if an error has occurred in the setting of the PLC link area FALSE: • if there are no transmission errors

7.7.6 PLC Link Response Time

The maximum value for the transmission time (T) of one cycle can be calculated using the following formula.

T max. = Ts1 + Ts2 + • • • • • + Tsn + Tlt + Tso + Tlk

① ② ③ ④

① Ts (transmission time per station)

Ts = scan time + Tpc

Tpc = Ttx × Pcm

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Pcm = 23 + (number of relay words + number of register words) × 4

Tpc (PLC link sending time)

Ttx (sending time per byte)

Pcm (PLC link sending size)

② Tlt (link table sending time)

Tlt = Ttx × Ltm

Ttx = 1/transmission speed ´ 1000 ´ 11ms » 0.096ms at 115200bit/s

Ltm = 13 + 2 × n

Ttx (sending time per byte)

Ltm (link table sending size)

n = number of stations being added

③ Tso (master station scan time)

The master station scan time should be confirmed using the programming tool.

④ Tlk (link addition processing time)

If no stations are being added, Tlk = 0.

Tlk = Tlc + Twt + Tls+ Tso

Tlc = 10 × Ttx

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Twt = Initial value 400ms (can be changed using SYS1 instruction)

Tls = 7 × Ttx

Ttx = 1/transmission speed × 1000 × 11ms ≈ 0.096ms at 115200bit/s

Tlc (link addition command sending time)

Twt (addition waiting time)

Ttx (sending time per byte)

Tls (link error stop command sending time)

Tso (master station scan time)

Ttx (sending time per byte)

Tso (master station scan time)

Calculation example 1

Conditions: All stations have been added to a 16-unit link. Highest station number = 16. Relays and registers have been evenly allocated. Scan time for each PLC: 1ms.

Ttx = 0.096

Pcm (per station) = 23 + (4 + 8) × 4 = 71

Tpc = Ttx × Pcm = 0.096 × 71 ≈ 6.82ms

Ts (per station) = 1 + 6.82 = 7.82ms

Tlt = 0.096 × (13 + 2 × 16) = 4.32ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 7.82 × 16 + 4.32 + 1 = 130.44ms

Calculation example 2

Conditions: All stations have been added to a 16-unit link. Highest station number = 16. Relays and registers have been evenly allocated. Relays and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (4 + 8) × 4 = 71

Tpc = Ttx × Pcm = 0.096 × 71 ≈ 6.82ms

Ts (per station) = 5 + 6.82 = 11.82ms

Tlt = 0.096 × (13 + 2 × 16) = 4.32ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 11.82 × 16 + 4.32 + 5 = 198.44ms

Calculation example 3

Conditions: All but one station have been added to a 16-unit link. Highest station number = 16. Relays and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Ts (per station) = 5 + 6.82 = 11.82ms

Tlt = 0.096 × (13 + 2 × 15) = 4.31ms

Tlk = 0.96 + 400 + 0.67 + 5 ≈ 407ms

Note: The default value for the addition waiting time is 400ms.

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 11.82 × 15 + 4.13 + 5 + 407 = 593.43ms

Calculation example 4

Conditions: All stations have been added to an 8-unit link. Highest station number = 8. Relays and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (8 + 16) × 4 = 119

Tpc = Ttx × Pcm = 0.096 × 119 ≈ 11.43ms

Ts (per station) = 5 + 11.43ms = 16.43ms

Tlt = 0.096 × (13 + 2 × 8) = 2.79ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 16.43 × 8 + 2.79 + 5 = 139.23ms

Calculation example 5

Conditions: All stations have been added to a 2-unit link. Highest station number = 2. Relays and registers have been evenly allocated. Scan time for each PLC: 5ms.

Ttx = 0.096

Pcm (per station) = 23 + (32 + 64) × 4 = 407

Tpc = Ttx × Pcm = 0.096 × 407 ≈ 39.072ms

Ts (per station) = 5 + 39.072 = 44.072ms

Tlt = 0.096 × (13 + 2 × 2) ≈ 1.632ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 44.072 × 2 + 1.632 + 5 = 94.776ms

Calculation example 6

Conditions: All stations have been added to a 2-unit link. Highest station number = 2. 32 relays and 2 register words have been evenly allocated. Scan time for each PLC: 1ms.

Ttx = 0.096

Pcm (per station) = 23 + (1 + 1) × 4 = 31

Tpc = Ttx × Pcm = 0.096 × 31 ≈ 2.976ms

Ts (per station) = 1 + 2.976 = 3.976ms

Tlt = 0.096 × (13 + 2 × 2) ≈ 1.632ms

Given the above conditions, the maximum value for the transmission time (T) of one cycle will be:

T max. = 3.976 × 2 + 1.632 + 1 = 10.584ms

◆NOTE

- **In the description, "stations that have been added" refers to stations which are connected between station no. 1 and the highest station number and for which the power supply has been turned on.**
- **Comparing examples 2 and 3, the transmission cycle time is longer if there is one station that has not been added to the link. As a result the PLC link response time is longer.**
- **The SYS1 instruction can be used to minimize the transmission cycle time even if there are one or more stations that have not been added to the link.**

7.7.6.1 Reducing Transmission Cycle Times

If there are stations that have not been added to the link, the link addition processing time (Tlk) and with this the transmission cycle time will be longer.

T max. = Ts1 + Ts2 + • • • • • • • + Tsn + Tlt + Tso + Tlk

Tlk = Tlc + Twt + Tls + Tso

Tlk= link addition processing time

Tlc = link addition command sending time

Twt = addition waiting time

Tls = link error stop command sending time

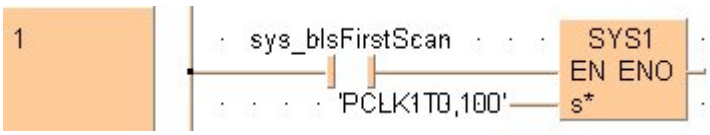
Tso = master station scan time

With the SYS1 instruction, the link addition waiting time (Twt) in the above formula can be reduced. Thus, SYS1 can be used to minimize the increase in the transmission cycle time.

◆EXAMPLE

Set SYS1 to change the waiting time for a link to be added to the PLC link from the default value of 400ms to 100ms.

LD Body



◆**NOTE**

- **If there are any stations that have not been added to the link, the setting should not be changed as long as a longer link transmission cycle time does not cause any problems.**
- **The SYS1 instruction should be executed at the beginning of the program, at the rise of R9014. The same waiting time should be set for all linked PLCs.**
- **The waiting time should be set to a value of at least twice the maximum scan time for any of the PLCs connected to the link.**
- **If a short waiting time has been set, there may be PLCs that cannot be added to the link even if their power supply is on. (The shortest time that can be set is 10ms.)**

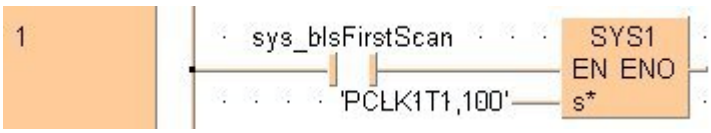
7.7.6.2 Error Detection Time for Transmission Assurance Relays

If the power supply of any given PLC fails or is turned off, it takes (as a default value) 6.4 seconds for the transmission assurance relay of that PLC to be turned off at the other stations. This time period can be shortened using the SYS1 instruction.

◆**EXAMPLE**

Set SYS1 to change the time that the PLC link transmission assurance is off from the default value of 6.4s to 100ms.

LD Body



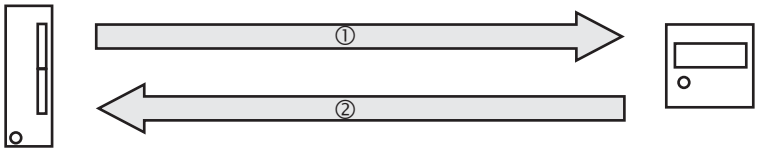
◆**NOTE**

- **The setting should not be changed as long as a longer transmission assurance relay detection time does not cause any problems.**

- **The SYS1 instruction should be executed at the beginning of the program, at the rise of R9014. The same waiting time should be set for all linked PLCs.**
- **The time should be set to a value of at least twice the maximum transmission cycle time when all of the PLCs are connected to the link.**
- **If a short time has been set, the transmission assurance relay may not function properly. (The shortest time that can be set is 100ms.)**

7.8 Modbus RTU Communication

The Modbus RTU protocol enables the communication between the FP0R and other devices (including the Panasonic FP-e PLCs, touch terminals of the GT series and KT temperature controllers). The master station sends instructions (command messages) to the slave stations and the slave stations respond (send response messages) based on the instructions received. The master station has read and write access to a maximum number of 99 slave stations.



①	Command message	②	Response message

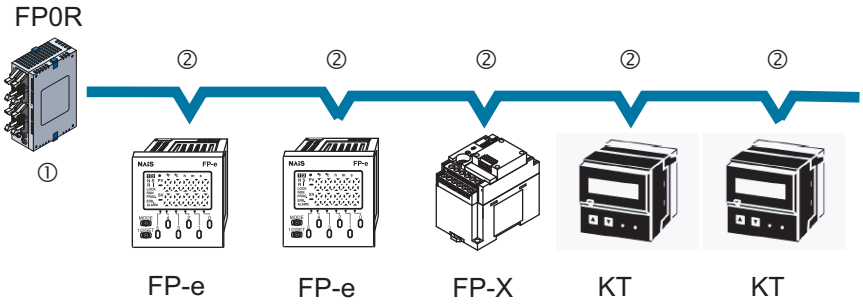
Modbus RTU connection between the FP0R and an external device

◆**NOTE**

The Modbus protocol supports both ASCII mode and RTU binary mode. However, the PLCs of the FP Series only support the RTU binary mode.

Modbus RTU master function

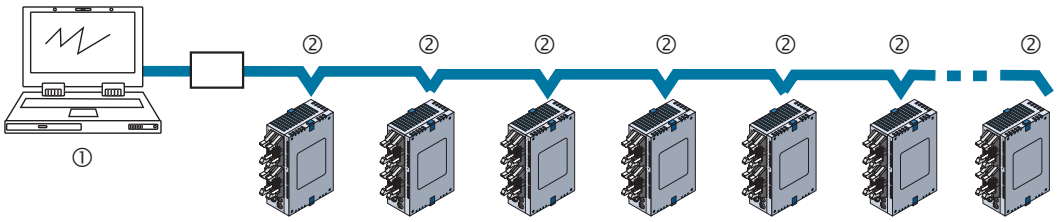
Write and read access to various slaves is possible using the F145_WRITE and F146_READ instructions. Individual access to each slave as well as global transmission is possible.



①	Master	②	Slave

Modbus RTU slave function

After having received a command message from the master station, the slave stations send back the response message based on the instructions received. Do not execute the F145_WRITE and F146_READ instructions on slave stations.



①	Master	②	Slave

Modbus RTU command message frame

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
Transmission time for 3.5 characters	8 bits	8 bits	n × 8 bits	16 bits	Transmission time for 3.5 characters

ADDRESS (station no.)	8 bits, 0–99 (decimal)[1)] 0 = broadcast address
FUNCTION	8 bits
DATA	Varies depending on the commands.
CRC	16 bits
END	Transmission time for 3.5 characters (differs depending on baud rate). Please refer to "Reception done judgment time".

[1)] FPWIN Pro does not support the address range from 0–247 of the Modbus RTU protocol.

Response in normal status

The same message as a command is returned for a single write command. A part of a command message (6 bytes from the beginning) is returned for a multiple write command.

Response in abnormal status

If a parameter which is to be processed but is disabled is found in a command (except for a transmission error):

ADDRESS	FUNCTION + 80H	ERROR CODE	CRC

ERROR CODE	1: Illegal function 2: Illegal data address (no word address) 3: Illegal data value (not a multiple of 16)

Reception done judgment time

The process for receiving a message is complete after all data has been received and the time given in this table has been reached.

Baud rate	Reception done judgment time
2400	≈13.3ms
4800	≈6.7ms
9600	≈3.3ms
19200	≈1.7ms
38400	≈0.8ms
57600	≈0.6ms
115200	≈0.3ms

Supported commands

Executable instructions for master	Code (decimal)	Name (Modbus original)	Name for FP0R	Modbus Reference
F146_READ	01	Read Coil Status	Read Y and R Coils	0X
F146_READ	02	Read Input Status	Read X Input	1X
F146_READ	03	Read Holding Registers	Read DT	4X
F146_READ	04	Read Input Registers	Read WL and LD	3X
F145_WRITE	05	Force Single Coil	Write Single Y and R	0X
F145_WRITE	06	Preset Single Register	Write DT 1 Word	4X
Cannot be issued	08	Diagnostics	Loopback Test	–
F145_WRITE	15	Force Multiple Coils	Write Multiple Ys and Rs	0X
F145_WRITE	16	Preset Multiple Registers	Write DT Multiple Words	4X
Cannot be issued	22	Mask Write 4X Register	Write DT Mask	4X
Cannot be issued	23	Read/Write 4X Registers	Read/Write DT	4X

Modbus references and FP0R addresses

Modbus reference				FP0R address
Name		Decimal address[1)]	Hexadecimal address[2)]	
Coil		000001–001760	0000–06DF	Y0–Y109F
		002049–006144	0800–17FF	R0–R255F
Input		100001–001760	0000–06DF	X0–X109F
Holding register	C10, C14, C16	400001–412316	0000–301B	DT0–DT12315
	C32, T32, F32	40001–432765	0000–7FFC	DT0–DT32765
Input register		300001–300128	0000–007F	WL0–WL127
		302001–302256	07D0–08CF	LD0–LD255

[1)] Beginning with 0 [2)] Beginning with 1

◆REFERENCE

For details on Modbus settings and communication using the F145_WRITE and F146_READ commands, please refer to the online help of FPWIN Pro.

7.8.1 Setting Communication Parameters

Make the following settings for the communication port:

- communication mode (MEWTOCOL-COM)
- station number
- baud rate
- communication format

For details on setting the communication parameters, see "Setting System Registers in PROG Mode" on page 94.

◆NOTE

- **The end code setting must always be "CR", and the start code setting must be "No STX".**
- **The station number can be set within a range of 1 to 99.**
- **With a C-NET adapter, a maximum of 32 stations can be specified.**

7.8.2 Sample Program for Master Communication

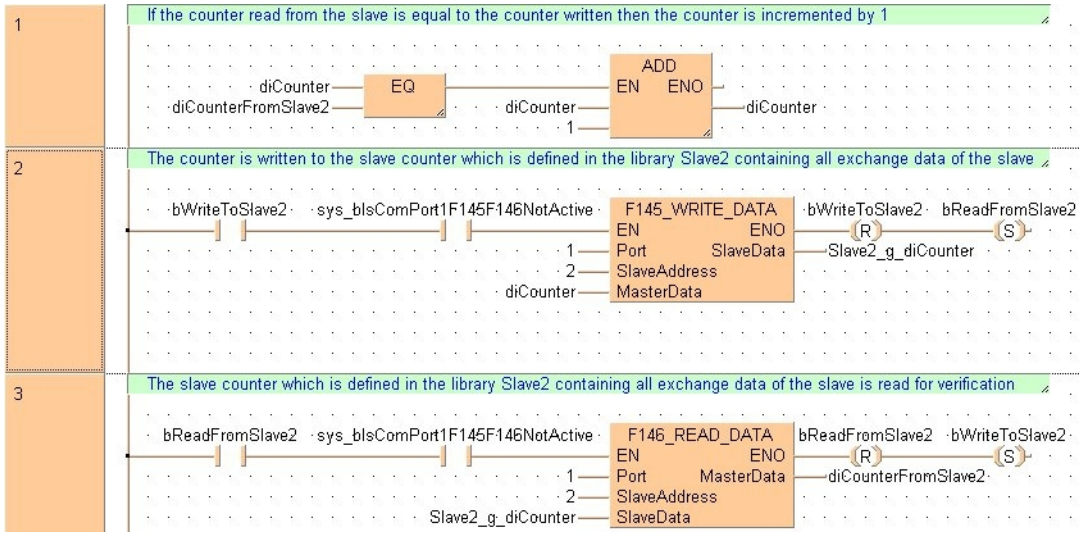
Use the F145_WRITE and F146_READ instructions for the Modbus master function. Be sure to select "Modbus RTU Master/Slave" for the COM port in system register 412.

POU Header

	Class	Identifier	Type	Initial
0	VAR_EXTERNAL	Slave2_g_diCounter	DINT	0
1	VAR	diCounter	DINT	0
2	VAR	diCounterFromSlave2	DINT	-1
3	VAR	bWriteToSlave2	BOOL	TRUE
4	VAR	bReadFromSlave2	BOOL	FALSE

In order to have consistent data in the master project and in the slave project, the common data should be kept in the GVL of a common library.

LD Body



ST Body

```
(* If the counter read from the slave is equal to the counter written
   then the counter is incremented by 1 *)
if (diCounter=diCounterFromSlave2) then
    diCounter:=diCounter+1;
end_if;

if (bWriteToSlave2 AND sys_bIsComPort1F145F146NotActive) then
    (* The counter is written to the slave counter
       which is defined in the library Slave2 containing all exchange data of the slave *)
    F145_WRITE_DATA(Port := 1,
                    SlaveAddress := 2,
                    MasterData := diCounter,
                    SlaveData => Slave2_g_diCounter;
    bWriteToSlave2:=FALSE;
    bReadFromSlave2:=TRUE;
end_if;

if (bReadFromSlave2 AND sys_bIsComPort1F145F146NotActive) then
    (* The slave counter
       which is defined in the library Slave2 containing all exchange data of the slave
       is read for verification *)
    F146_READ_DATA(Port := 1,
              SlaveAddress := 2,
              SlaveData := Slave2_g_diCounter,
              MasterData=> diCounterFromSlave2);
    bReadFromSlave2:=FALSE;
    bWriteToSlave2:=TRUE;
end_if;
```

◆REFERENCE

For details on Modbus settings and communication using the F145_WRITE and F146_READ commands, please refer to the online help of FPWIN Pro.

Chapter 8

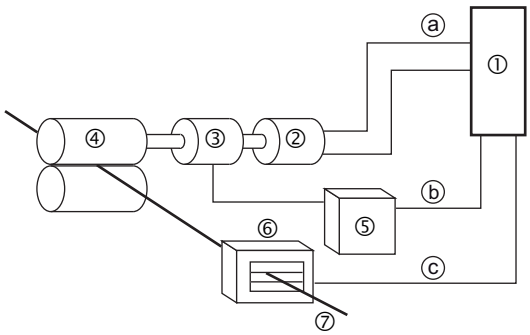
High-Speed Counter and Pulse Output

8.1 Overview

Three built-in hardware extensions allow the FP0R to be used for positioning control and measurement: high-speed counting, pulse output, and PWM (pulse-width modulation) output.

High-speed counter function

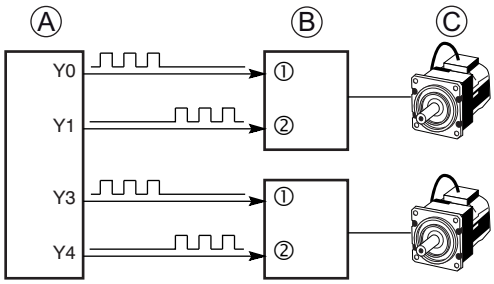
The high-speed counter function counts external inputs such as those from sensors or encoders. When the count reaches the target value, this function turns the desired output to TRUE or to FALSE.



①	PLC		
②	Encoder	ⓐ	Encoder output is input to the high-speed counter
③	Motor		
④	Roller		
⑤	Inverter	ⓑ	Start/stop signal
⑥	Cutter	ⓒ	Cutter blade control signal
⑦	Tape, lead wire		

Pulse output function

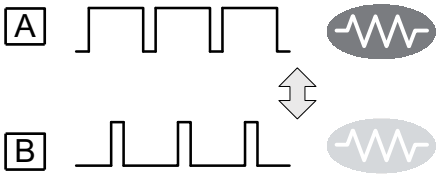
By connecting a commercially available motor driver to the PLC, positioning control can be performed with the pulse output function. Using special instructions, trapezoidal control, home return, or JOG operation is possible.



Ⓐ	PLC	①	CW pulse output
Ⓑ	Motor driver	②	CCW pulse output
Ⓒ	Stepping motor/servo motor		

PWM output function

A special instruction makes it possible to output pulses with a specified duty ratio.



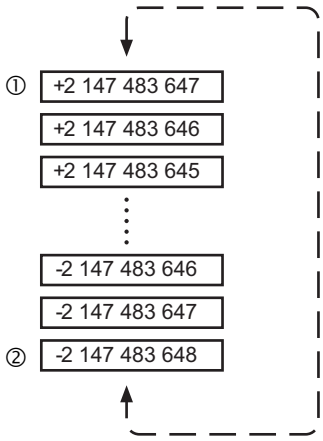
Heater control using the PWM output function

①	Increasing the pulse width increases heating
②	Decreasing the pulse width decreases heating

Counting Range

The counting range of the built-in high-speed counter is from -2 147 483 648 to 2 147 483 647 (32-bit binary number).

The high-speed counter is a ring counter. Consequently, if the counted value exceeds the maximum value, the counter returns to the minimum value. Similarly, if the counted value drops below the minimum value, the counter goes back to the maximum value and continues counting from there.



①	Maximum value
②	Minimum value

◆**NOTE**

Using linear interpolation instruction F175_PulseOutput_Linear: The target value or the amount of travel must be within the range of -8 388 608 to +8 388 607 (24-bit binary number).

8.2 Function Specifications and Restrictions

This section contains the specifications and restrictions of the high-speed counter, pulse output, and PWM output function.

8.2.1 High-Speed Counter Function

For each count input mode, there are certain high-speed counter channels, inputs, and memory areas.

Input numbers

Input mode [1]	No. of phases	Channel no.	Input	Reset input [2]
• Incremental • Decremental	1	0	X0	X2
		1	X1	X2
		2	X3	X5
		3	X4	X5
		4[3]	X6	–
		5[3]	X7	–
• Two-phase • Incremental/decremental • Incremental/decremental control	2	0	X0, X1	X2
		2	X3, X4	X5
		4[3]	X6, X7	–

1) For details on the different input modes, see page 166.

2) Reset input X2 can be set to either channel 0 or channel 1. Reset input X5 can be set to either channel 2 or channel 3.

3) Channel 4 and channel 5 are not available for the C10 type.

Performance

No. of phases	Minimum input pulse width [1]	No. of channels	Maximum counting speed [2]
1	10μs	5	50kHz
2	25μs	1	15kHz
		2	15kHz (×2 channels)
		3	10kHz (×3 channels)

1) For information on the minimum input pulse width, see page 168.

2) The maximum counting speed may be lower than the values indicated in the table when the pulse output speed is changed, or when a cam control, target value match on/off, or other interrupt program is executed simultaneously.

Control flags and memory areas

The high-speed counter operating status, counting values, and control code are stored in special internal relays and special data registers. The control code contains the counter settings. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. See "Instructions and System Variables" on page 169.

Related instructions

F165_HighSpeedCounter_Cam: Cam control

F166_HighSpeedCounter_Set: Target value match ON

F167_HighSpeedCounter_Reset: Target value match OFF

F178_HighSpeedCounter_Measure: Input pulse measurement

8.2.2 Pulse Output Function

For each pulse output mode and position control mode there are certain designated high-speed counter channels, inputs and outputs.

◆NOTE

The pulse output function is only available with the transistor output type.

Input/output numbers

Channel no.			CW pulse output	CCW pulse output	Deviation counter clear output [1]	Home input	Near home input
			Pulse output	Direction output			
0			Y0	Y1	Y6 (Y8)	X4	Any[3]
1			Y2	Y3	Y7 (Y9)	X5	
2			Y4	Y5	– (YB)	X6	
3			Y6	Y7	YB	X7	
Linear interpolation control [2]	0	X axis	Y0	Y1	Y6 (Y8)	X4	
		Y axis	Y2	Y3	Y7 (Y9)	X5	
	1	X axis	Y4	Y5	– (YA)	X6	
		Y axis	Y6	Y7	– (YB)	X7	

1) The values in parentheses refer to the CPU type C32.

2) The home return operation of the interpolation axes should be performed for every channel.

3) Any input can be specified in the global variable list. The near home input is enabled/disabled using the pulse output control code. See "Writing the Pulse Output Control Code" on page 185.

Performance

No. of channels	Maximum output frequency1)
4	50kHz
Linear interpolation control	50kHz

1) The maximum output frequency may be lower than the values indicated in the table when the pulse output speed is changed, when a target value match on/off instruction, another pulse I/O process or interrupt program is executed simultaneously.

Control flags and memory areas

Counter and pulse output settings as well as elapsed values are stored in special data

registers. The pulse output status is stored in special internal relays. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. See "Instructions and System Variables" on page 183.

Related instructions

F166_PulseOutput_Set: Target value match ON (pulse output)

F167_PulseOutput_Reset: Target value match OFF (pulse output)

F171_PulseOutput_Trapezoidal: Trapezoidal control

F171_PulseOutput_Jog_Positioning: JOG operation and positioning

F172_PulseOutput_Jog: JOG operation

F174_PulseOutput_DataTable: Data table control

F175_PulseOutput_Linear: Linear interpolation control

F177_PulseOutput_Home: Home return

8.2.3 PWM Output Function

For the pulse-width modulation output function there are two designated channels and outputs.

◆**NOTE**

The PWM output function is only available with the transistor output type.

Output numbers

Channel no.	PWM output
0	Y0
1	Y2
2	Y4
3	Y6

Performance

Resolution	Output frequency (duty ratio)
1000	6Hz–4.8kHz (0.0–99.9%)

Control flags

The PWM output status is stored in special internal relays. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. See "PWM Output Function" on page 198.

Related instructions

F173_PWMH: PWM output

8.2.4 Maximum Counting Speed and Output Frequency

The maximum counting speed of the high-speed counter function is determined by the number of channels used and the simultaneous use of the pulse output function. Use the following simplified chart as a guide.

◆NOTE

The maximum counting speed may be lower than the values indicated in the table when the pulse output speed is changed, or when a cam control, target value match on/off, or other interrupt program is executed simultaneously.

Maximum counting speed

No.[1]	Combination of high-speed counter channels									Maximum counting speed (frequency) [kHz][2]			
										No pulse output		Pulse output, 1 channel	
	1-phase						2-phase			1-phase	2-phase	1-phase	2-phase
	Channel						Channel						
	0	1	2	3	4	5	0	2	4				
1	X									50		50	
2	X	X								50		50	
3	X	X	X							50		50	
4	X	X	X	X						50		50	
5	X	X	X	X	X					50		40	
6	X	X	X	X	X	X				50		40	
7							X				15		14
8							X	X			15		10
9							X	X	X		10		10
10			X				X			50	15	50	14
11			X	X			X			50	15	50	14
12			X	X	X		X			50	15	50	14
13			X	X	X	X	X			50	15	50	14
14					X		X	X		50	15	50	10
15					X	X	X	X		50	15	50	10
16	X								X	50	15	50	12
17	X	X							X	50	13	50	12
18	X	X	X						X	50	12	50	11
19	X	X	X	X					X	50	12	50	9
20	X							X	X	50	13	50	10
21	X	X						X	X	50	12	50	10

X:	Channel is being used
1)	The numbers are reference numbers for the specifications continued in the next table.
2)	When combined with pulse output function: trapezoidal control, no change in speed (50kHz)

(Table continued)

Please use the reference numbers to read the table.

No.	Maximum counting speed [kHz][1)]					
	Pulse output, 2 channels		Pulse output, 3 channels		Pulse output, 4 channels	
	1-phase	2-phase	1-phase	2-phase	1-phase	2-phase
1	50		50		30	
2	50		35		25	
3	50		30		20	
4	40		30		20	
5	35		29		20	
6	30		24		15	
7		10		10		10
8		9		8		8
9		9		8		8
10	50	10	44	10	30	10
11	50	10	40	10	28	10
12	44	10	30	10	25	10
13	35	10	25	10	20	10
14	50	9	35	8	28	8
15	40	9	30	8	25	8
16	50	10	50	10	40	8
17	50	10	45	8	35	7
18	50	9	40	8	30	7
19	50	8	35	8	30	7
20	50	10	50	8	40	8
21	50	9	45	8	35	7

1) When combined with pulse output function: trapezoidal control, no change in speed (50kHz)

Maximum output frequency

◆NOTE

The maximum output frequency may be lower than the values indicated in the table when the pulse output speed is changed, when a target value match on/off instruction, another pulse I/O process or interrupt program is executed simultaneously.

Using channels independently: Even if all channels are used, the maximum output frequency is 50kHz for all.

1-phase				Maximum output frequency [kHz]
Channel 0	Channel 1	Channel 2	Channel 3	
X				50
X	X			50
X	X	X		50
X	X	X	X	50

X: Channel is being used

Using linear interpolation control: Even if all channels are used for interpolation, the maximum output frequency is 50kHz for all.

Linear interpolation control		Maximum output frequency [kHz]
Channel 0	Channel 2	
X		50
X	X	50

X: Channel is being used

8.3 High-Speed Counter Function

The high-speed counter function counts the input signals and sets the desired output to TRUE or to FALSE when the target value is reached. The high-speed counter function can also be used for cam control and for input pulse measurement.

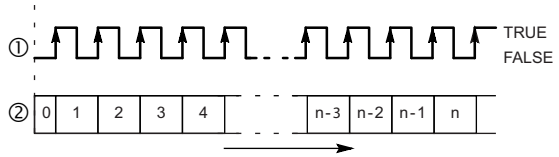
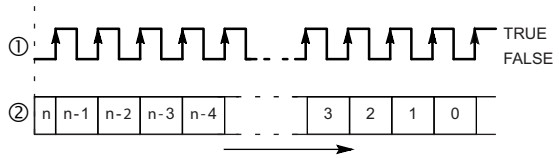
Setting the system registers

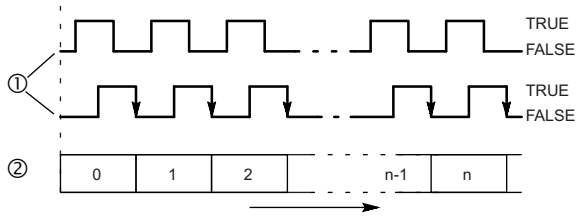
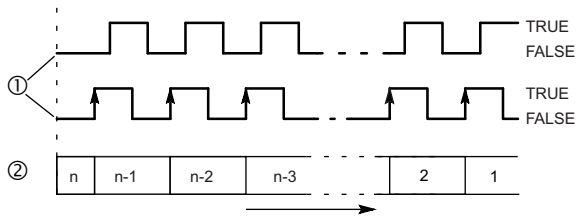
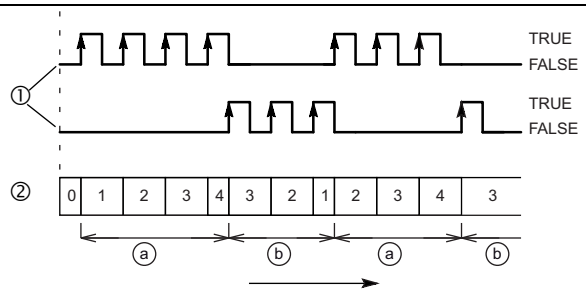
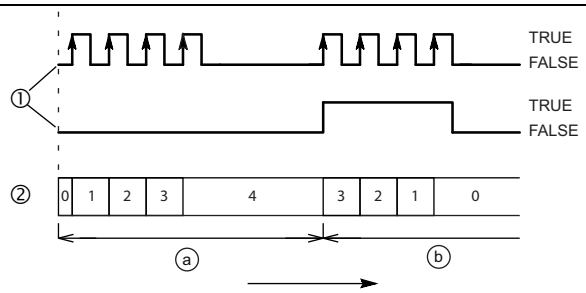
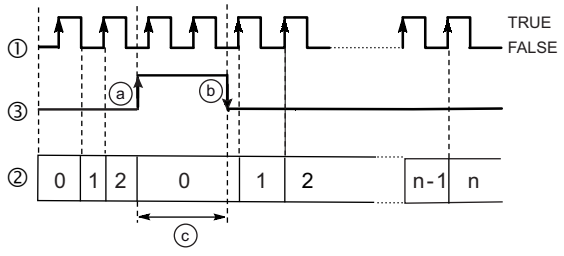
In order to use the high-speed counter function, it is necessary to set the inputs in the system registers.

◆Procedure

1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **Double-click "High-Speed Counter, Pulse-Catch Input, Interrupt Input"**
4. **Select the desired inputs for each channel**

8.3.1 Count Input Modes

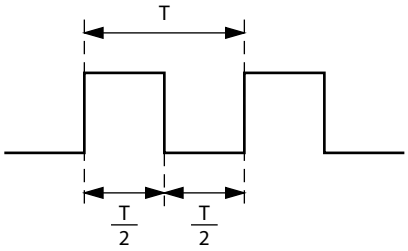
Input mode	Input signals	
Incremental		① Input: X0 (X1, X3, X4, X6, X7) ② Count
Decremental		

<table>
<tr><th>Input mode</th><th>Input signals</th><th></th></tr>
<tr><td rowspan="2">Two-phase</td><td>Incremental input: CW
 
</td><td rowspan="2">① Inputs: X0+X1 (X3+X4 or X4+X7)
② Count</td></tr>
<tr><td>Decremental input: CCW
 
</td></tr>
<tr><td>Incremental/decremental</td><td> 
</td><td>① Inputs: X0+X1 (X3+X4 or X4+X7)
② Count
ⓐ Increasing
ⓑ Decreasing</td></tr>
<tr><td>Incremental/decremental control</td><td> 
</td><td>① Inputs: X0+X1 (X3+X4 or X4+X7)
② Count
ⓐ Increasing
ⓑ Decreasing</td></tr>
<tr><td>Count for reset (incremental)</td><td> 

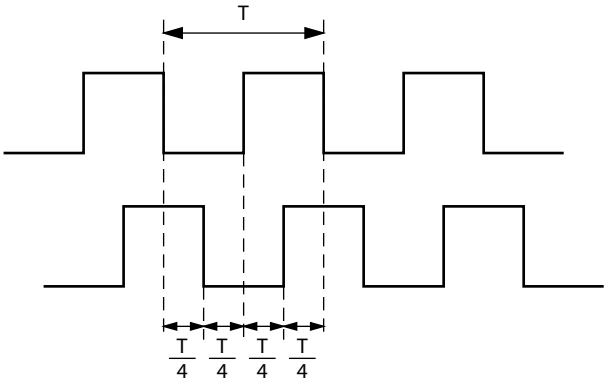
The reset is executed by the interruption at ⓐ (rising edge) and ⓑ (falling edge)
The reset input can be enabled/disabled using bit 2 of sys_wHscOrPulseControlCode (see page 170).</td><td>① Inputs: X0 or X1 (X3 or X4)
② Reset input: X2 (X5)
③ Count
ⓐ Rising edge: count disabled, elapsed value cleared
ⓑ Falling edge: count enabled
ⓒ Count prohibited</td></tr>
</table>

8.3.2 Minimum Input Pulse Width

For the period T (1/frequency), a minimum input pulse width of T/2 (single-phase input) or T/4 (two-phase input) is required.



Single-phase input



Two-phase input

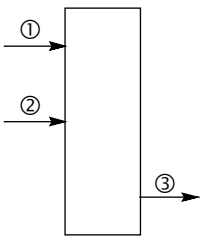
8.3.3 I/O Allocation

The inputs and outputs used will differ depending on the channel number being used. (See "Function Specifications and Restrictions" on page 160.)

The output to be turned to TRUE or to FALSE can be specified with the instructions **F166_HighSpeedCounter_Set** and **F167_HighSpeedCounter_Reset**. Outputs can be specified from Y0 to Y7.

Using channel 0 with incremental input and reset input

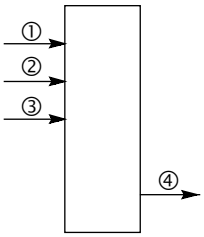
FP0R:



①	Count input X0
②	Reset input X2
③	TRUE/FALSE output
Yn	The output which is turned TRUE or FALSE when the target value is reached: Y0–Y7

Using channel 0 with two-phase input and reset input

FP0R:



①	Phase A input X0
②	Phase B input X1
③	Reset input X2
④	TRUE/FALSE output
Yn	The output which is turned TRUE or FALSE when the target value is reached: Y0–Y7

8.3.4 Instructions and System Variables

The instruction **F165_HighSpeedCounter_Cam** performs Cam control according to the parameters in the specified DUT.

Use the instructions **F166_HighSpeedCounter_Set** or **F167_HighSpeedCounter_Reset** to turn the desired output to TRUE or to FALSE when the specified target value is reached. To turn the output to TRUE, use **F166_HighSpeedCounter_Set**. To turn the output to FALSE, use **F167_HighSpeedCounter_Reset**.

The instruction **F178_HighSpeedCounter_Measure** measures the number of input pulses in a specified counting period and the pulse period.

The high-speed counter operating status, counting values, and control code are stored in special internal relays and special data registers. The control code contains the counter settings. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. Please refer to the FPWIN Pro online help for detailed information on using system variables.

System variables for memory areas used

Description		System variable	Address
High-speed counter: control flag for channel	0	sys_bIsHscChannel0ControlActive	R9110
	1	sys_bIsHscChannel1ControlActive	R9111
	2	sys_bIsHscChannel2ControlActive	R9112
	3	sys_bIsHscChannel3ControlActive	R9113
	4	sys_bIsHscChannel4ControlActive	R9114
	5	sys_bIsHscChannel5ControlActive	R9115
High-speed counter: elapsed value of channel	0	sys_diHscChannel0ElapsedValue	DDT90300
	1	sys_diHscChannel1ElapsedValue	DDT90304
	2	sys_diHscChannel2ElapsedValue	DDT90308
	3	sys_diHscChannel3ElapsedValue	DDT90312
	4	sys_diHscChannel4ElapsedValue	DDT90316
	5	sys_diHscChannel5ElapsedValue	DDT90320
High-speed counter: target value of channel	0	sys_diHscChannel0ControlTargetValue	DDT90302
	1	sys_diHscChannel1ControlTargetValue	DDT90306
	2	sys_diHscChannel2ControlTargetValue	DDT90310
	3	sys_diHscChannel3ControlTargetValue	DDT90314
	4	sys_diHscChannel4ControlTargetValue	DDT90318
	5	sys_diHscChannel5ControlTargetValue	DDT90322
High-speed counter: control code monitor for channel	0	sys_wHscChannel0ControlCode	DT90370
	1	sys_wHscChannel1ControlCode	DT90371
	2	sys_wHscChannel2ControlCode	DT90372
	3	sys_wHscChannel3ControlCode	DT90373
	4	sys_wHscChannel4ControlCode	DT90374
	5	sys_wHscChannel5ControlCode	DT90375
High-speed counter or pulse output control code		sys_wHscOrPulseControlCode	DT90052

8.3.4.1 Writing the High-Speed Counter Control Code

The special data register where the high-speed counter and pulse output control code are stored can be accessed with the system variable sys_wHscOrPulseControlCode. (The system variable sys_wHscOrPulseControlCode corresponds to special data register DT90052.)

The control code settings for each channel can be monitored using the system variables sys_wHscChannelxControlCode or sys_wPulseChannelxControlCode (where x=channel number). The settings of this system variable remain unchanged until another setting operation is executed.

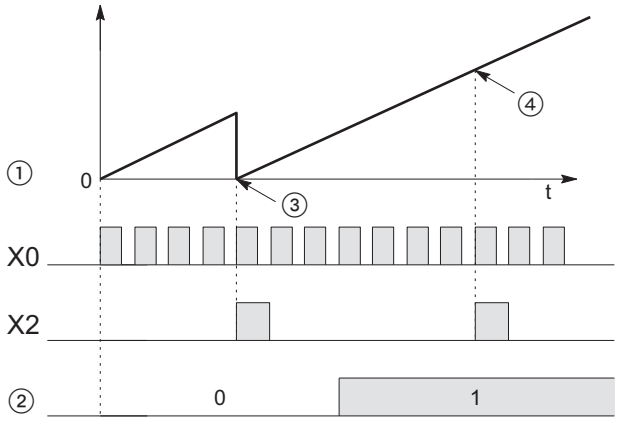
Operations performed by the high-speed counter control code:

- Clearing high-speed counter instructions (bit 3)
- Enabling/disabling the reset input (hardware reset) of the high-speed counter (bit 2)
- Enabling/disabling counting operations (bit 1)
- Resetting the elapsed value (software reset) of the high-speed counter to 0 (bit 0)

Clearing high-speed counter instructions (bit 3)

To cancel execution of an instruction, set bit 3 of the data register storing the high-speed counter control code (sys_wHscOrPulseControlCode) to TRUE. The high-speed counter control flag then changes to FALSE. To reenable execution of the high-speed counter instruction, reset bit 3 to FALSE.

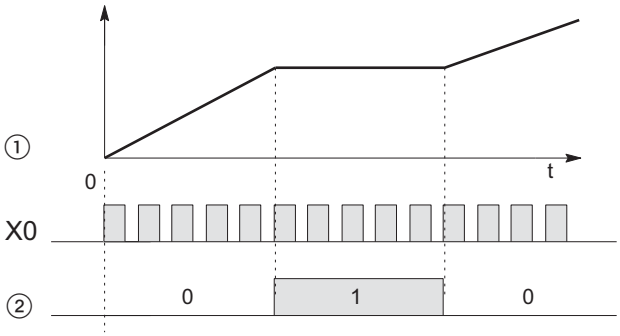
Enabling/disabling the reset input (hardware reset) of the high-speed counter (bit 2)



X0	High-speed counter input
①	Elapsed value
②	Bit 2 of high-speed counter control code (enable/disable reset input)
③	Elapsed value is reset to 0
④	Reset not possible

When bit 2 of the control code is set to TRUE, a hardware reset using the reset input specified in the system registers is not possible. Counting will continue even if the reset input has turned to TRUE. The hardware reset is disabled until bit 2 is reset to 0.

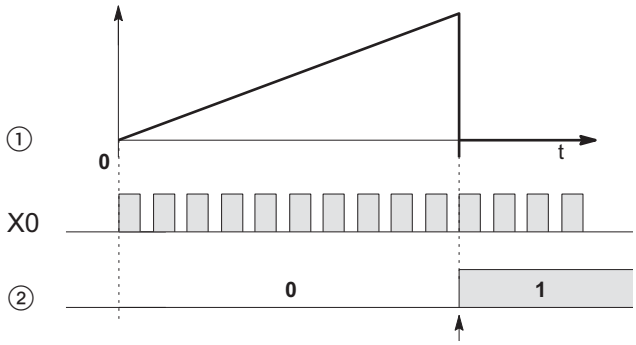
Enabling/disabling counting operations (bit 1)



X0	High-speed counter input
①	Elapsed value
②	Bit 1 of high-speed counter control code (count)

When bit 1 of the control code is set to TRUE, counting is prohibited and the elapsed value keeps its current value. Counting is continued when bit 1 is reset to FALSE.

Resetting the elapsed value (software reset) of the high-speed counter to 0 (bit 0)

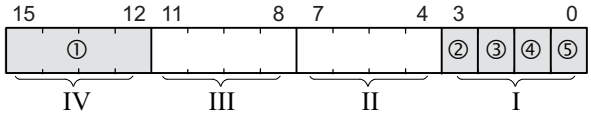


X0	High-speed counter input
①	Elapsed value
②	Bit 0 of high-speed counter control code (software reset)

When bit 0 of the control code is set to TRUE, a software reset is performed and the elapsed value is set to 0. The elapsed value keeps the value 0 until bit 0 is reset to FALSE.

Control code settings

Bits 0–15 of the control code are allocated in groups of four. The bit setting in each group is represented by a hex number (e.g. 0002 0000 0000 1001 = 16#2009).



Group IV	①	Channel number (channel n: 16#n)	
Group III		0 (fixed)	
Group II		0 (fixed)	
Group I	②	Clear high-speed counter instruction (bit 3)	
		0: continue	1: clear
	③	Reset input (bit 2) (see note)	
		0: enabled	1: disabled
	④	Count (bit 1)	
		0: permit	1: prohibit
	⑤	Reset elapsed value to 0 (bit 0)	
		0: no	1: yes

Example: 16#2009

Group	Value	Description	
IV	2	Channel number: 2	
III	0	(fixed)	
II	0	(fixed)	
I	9	Hex 9 corresponds to binary 1001	
		Clear high-speed counter instruction: clear (bit 3)	1
		Reset input: enabled (bit 2)	0
		Count: permit (bit 1)	0
		Reset elapsed value to 0: yes (bit 0)	1

◆**NOTE**

Use the reset input setting (bit 2) to disable the reset input allocated in the system registers.

◆REFERENCE

Please refer to the FPWIN Pro online help for programming examples.

8.3.4.2 Writing and Reading the Elapsed Value for the High-Speed Counter

The elapsed value is stored as a double word in the special data registers. Access the special data registers using the system variable sys_diHscChannelxElapsedValue (where x=channel number).

System variables for memory areas used:

Description		System variable	Address
High-speed counter: elapsed value of channel	0	sys_diHscChannel0ElapsedValue	DDT90300
	1	sys_diHscChannel1ElapsedValue	DDT90304
	2	sys_diHscChannel2ElapsedValue	DDT90308
	3	sys_diHscChannel3ElapsedValue	DDT90312
	4	sys_diHscChannel4ElapsedValue	DDT90316
	5	sys_diHscChannel5ElapsedValue	DDT90320

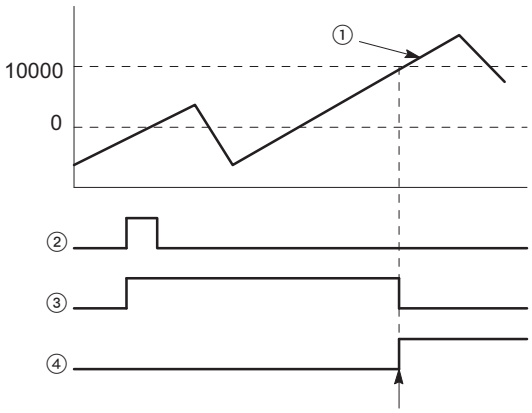
◆REFERENCE

Please refer to the FPWIN Pro online help for programming examples.

8.3.4.3 F166_HighSpeedCounter_Set, Target Value Match ON

If the elapsed value of the high-speed counter matches the target value, an interrupt process immediately turns the specified output to TRUE.

Characteristics of target value match ON control



10000	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	High-speed counter control flag
④	PLC output

The PLC output turns to TRUE when the elapsed value matches the target value. In addition, the high-speed counter control flag turns to FALSE and the instruction is deactivated.

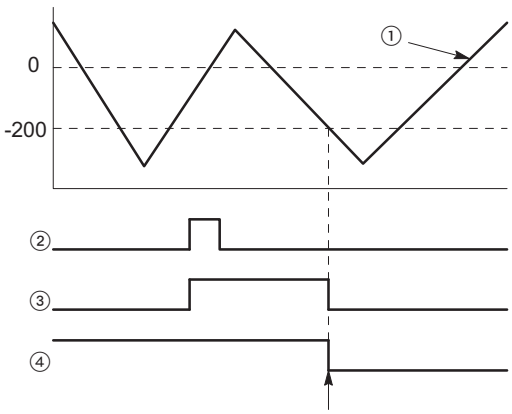
◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.3.4.4 F167_HighSpeedCounter_Reset, Target Value Match OFF

If the elapsed value of the high-speed counter matches the target value, an interrupt process immediately turns the specified output to FALSE.

Characteristics of target value match OFF control



-200	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	High-speed counter control flag
④	PLC output

The PLC output turns to FALSE when the elapsed value matches the target value. In addition, the high-speed counter control flag turns to FALSE and the instruction is deactivated.

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.3.4.5 F178_HighSpeedCounter_Measure, Input Pulse Measurement

This instruction measures the number of input pulses in a specified counting period and the pulse period.

Characteristics of input pulse measurement

- For input pulse measurement, the channel number, the counting period (1ms–5s) and the number of counting periods (1–5) must be specified. These parameters are used to calculate the average number of input pulses per counting period.
- The unit of pulse period measurement ([μs], [ms] or both) can be specified.

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.3.5 Sample Programs

The following programming examples demonstrate how to make control code settings and how to use the high-speed counter instructions.

The FPWIN Pro projects in LD and ST code can be downloaded from the Panasonic Web site (http://www.panasonic-electric-works.com/peweu/en/html/22164.php).

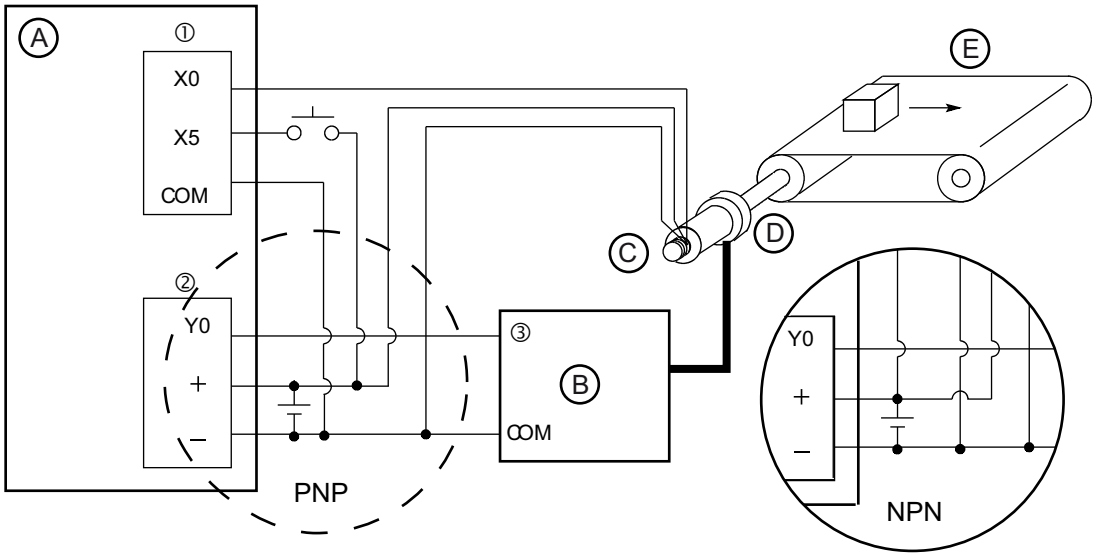
The programming examples for this chapter can be found in pe_63403_0001_sample_high_speed.zip.

These examples can be used with different PLC types. Therefore you have to adapt the PLC type in the FPWIN Pro Navigator.

After you have changed the PLC type, a message appears: "Adapt System Registers and Compile Options?" Select [Adapt automatically], so that you do not lose the system register settings set in the programming example.

8.3.5.1 Positioning Operations with a Single-Speed Inverter

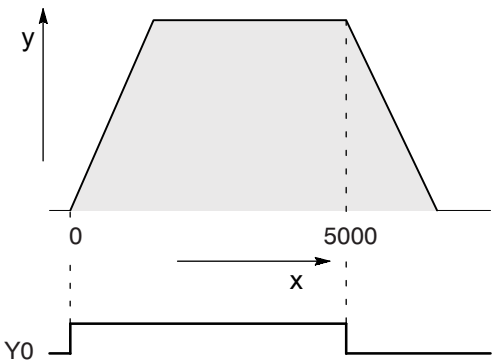
Wiring example



Ⓐ	PLC	①	Input terminal	X0	Encoder input	X5	Operation start
		②	Output terminal	Y0	Inverter operation		
Ⓑ	Inverter	③	Operation/stop				
Ⓒ	Encoder						
Ⓓ	Motor						
Ⓔ	Conveyor						

When X5 turns to TRUE, Y0 turns to TRUE and the conveyor starts to move. When the elapsed value (sys_diHscChannel0ElapsedValue) reaches 5000, Y0 turns to FALSE and the conveyor stops.

Operation chart



x	Number of pulses
y	Speed

System register settings

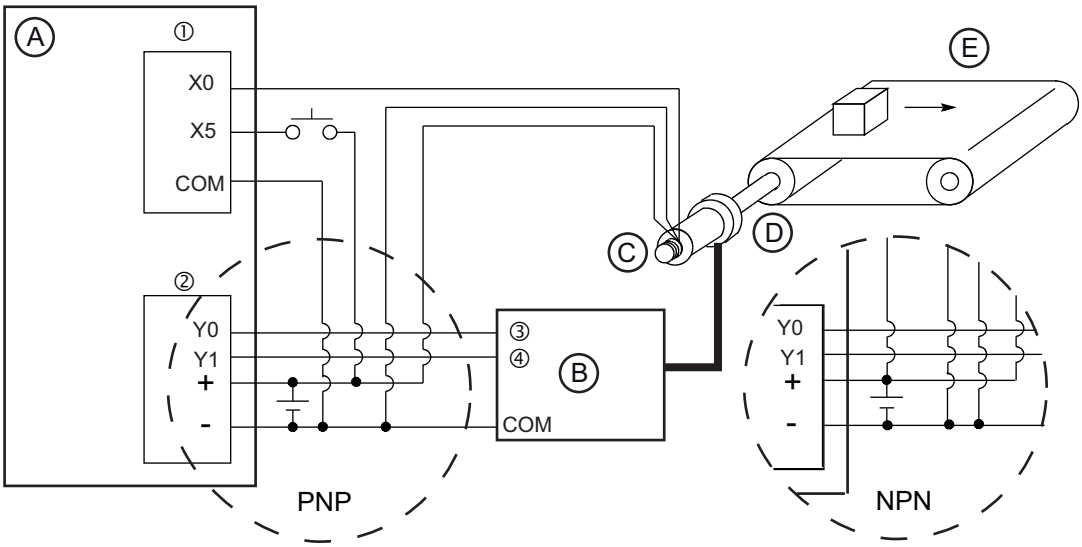
No	Item Name	Data
400	High-speed counter: Channel 0	Incremental input (X0)

◆REFERENCE

For POU Header and POU Body, please see the programming examples in Panasonic's download area.

8.3.5.2 Positioning Operations with a Double-Speed Inverter

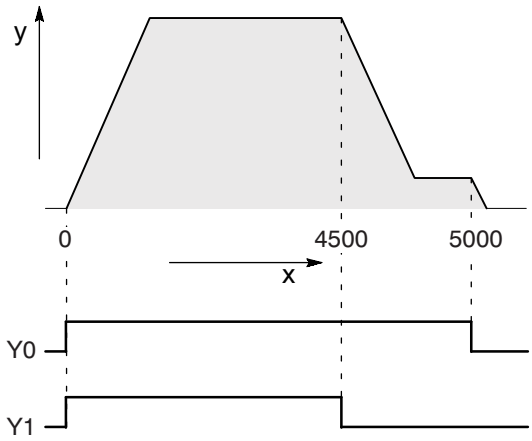
Wiring example



Ⓐ	PLC	①	Input terminal	X0	Encoder input	X5	Operation start
		②	Output terminal	Y0	Inverter operation	Y1	Inverter high-speed
Ⓑ	Inverter	③	Operation/stop				
		④	Fast/slow				
Ⓒ	Encoder						
Ⓓ	Motorr						
Ⓔ	Conveyor						

When X5 turns to TRUE, Y0 and Y1 turn to TRUE and the conveyor begins to move. When the elapsed value (sys_diHscChannel0ElapsedValue) reaches 4500, Y1 turns to FALSE and the conveyor begins to decelerate. When the elapsed value reaches 5000, Y0 turns to FALSE and the conveyor stops.

Operation chart



x	Number of pulses
y	Speed

System register settings

No	Item Name	Data
400	High-speed counter: Channel 0	Incremental input (X0)

◆REFERENCE

For POU Header and POU Body, please see the programming examples in Panasonic's download area.

8.4 Pulse Output Function

Together with a commercially available pulse-string input type motor driver, the pulse output function can be used for positioning control.

◆**NOTE**

The pulse output function is only available with the transistor output type.

Setting system registers

When using the pulse output function, make sure the high-speed counter function is not allocated to the channel selected for pulse output.

◆**Procedure**

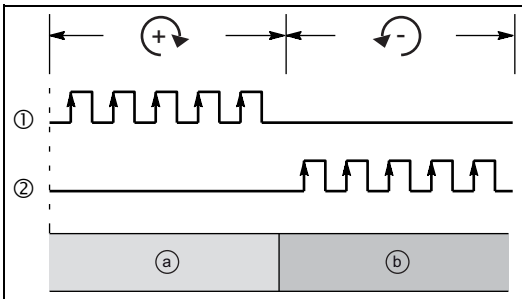
1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **Double-click "High-Speed Counter, Pulse-Catch Input, Interrupt Input"**
4. **Set any high-speed counter allocated to a pulse output channel to "Unused"**

No	Item Name	Data	Dime...	Range
400	High-speed counter: Channel 0	Unused		Unused
400	High-speed counter: Channel 1	Unused		Unused
401	High-speed counter: Channel 2	Unused		Unused
401	High-speed counter: Channel 3	Unused		Unused

8.4.1 Pulse Output Methods and Position Control Modes

The pulse output method and position control mode are specified by means of the variables used with the positioning command.

CW/CCW

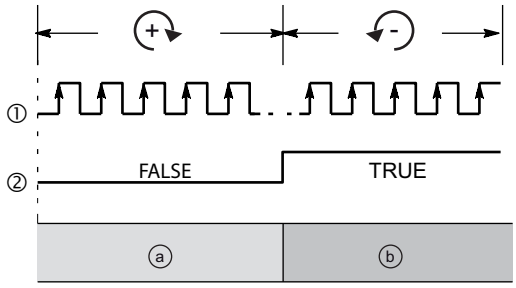


Control is carried out using two pulses: a positive or clockwise rotation pulse (CW) and a negative or counterclockwise rotation pulse (CCW pulse).

①	CW pulse output: Y0 (Y2)
②	CCW pulse output Y1 (Y3)
ⓐ	Incremental counting
ⓑ	Decremental counting

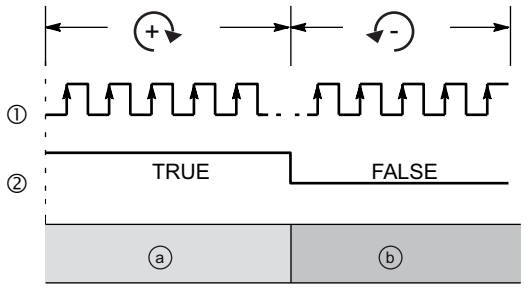
Pulse/direction

Forward FALSE



Control is carried out using one pulse output to specify the speed and another to specify the direction of rotation with TRUE/FALSE signals. In this mode, forward rotation is carried out when the rotation direction signal is FALSE.

Forward TRUE



Control is carried out using one pulse output to specify the speed and another to specify the direction of rotation with TRUE/FALSE signals. In this mode, forward rotation is carried out when the rotation direction signal is TRUE.

① Pulse output: Y0 (Y2)

② Direction output: Y1 (Y3)

ⓐ Incremental counting

ⓑ Decremental counting

Relative value control

The number of pulses set with the target value is output. Positive values result in a positive rotation, negative values in a negative rotation.

◆EXAMPLE

With a current position of 5000 and a target value of +1000, 1000 pulses are output from CW to reach the new position at 6000.

Absolute value control

A number of pulses equal to the difference between the set target value and the current value are output. Values greater than the current value result in a positive rotation, values smaller than the current value result in a negative rotation.

◆EXAMPLE

With a current position of 5000 and a target value of +1000, 4000 pulses are output from CCW to reach the new position at 1000.

The following outputs are TRUE or FALSE depending on the selected pulse output method and position control mode:

Pulse output method		Pulse output	Target value	
			Positive value/ > current value	Negative value/ < current value
CW/CCW		CW	TRUE	FALSE
		CCW	FALSE	TRUE
Pulse/direction	Forward FALSE	Pulse	TRUE	TRUE
		Direction	FALSE	TRUE
	Forward TRUE	Pulse	TRUE	TRUE
		Direction	TRUE	FALSE
Count mode			Incremental counting	Decremental counting

Home return

After a drive system has been switched on, there is a difference between the internal position value (elapsed value) and the mechanical position of the axis; this difference cannot be predetermined. The internal value must be synchronized with the actual position value of the axis. This is done by means of a home return, during which a position value is registered at a known reference point (home).

During execution of a home return instruction, pulses are continuously output until the home input is enabled. The I/O allocation is determined by the channel used. See "I/O Allocation" on page 182.

To decelerate movement when near the home position, designate a near home input and set bit 4 of the special data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE and back to FALSE again.

The deviation counter clear output can be set to TRUE when home return has been completed.

JOG operation

Pulses are output from the specified channel while the trigger for the **F172_PulseOutput_Jog** instruction is TRUE. Direction output and output frequency are specified with this instruction.

8.4.2 I/O Allocation

The I/O allocation of pulse output terminals, direction output terminal, and home input is determined by the channel used.

For the near home input, the desired contact must be allocated and bit 4 of the special data register storing the pulse output control code (sys_wHscOrPulseControlCode) must be set to TRUE and back to FALSE again.

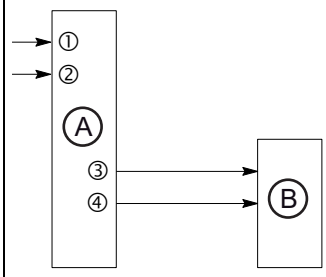
◆REFERENCE

The input/output numbers are indicated by channel in the specifications. See "Pulse Output Function" on page 161.

Double pulse input driver (CW/CCW pulse output method)

Two output contacts are used as a pulse output for CW/CCW.

Set the control code for **F171** to CW/CCW.



Using channel		0	2
Ⓐ	PLC		
Ⓑ	Motor driver		
①	Home input	X4	X6
②	Near home input (see note)	e.g. X0	e.g. X1
③	CW pulse output	Y0	Y4
④	CCW pulse output	Y1	Y5

◆**NOTE**

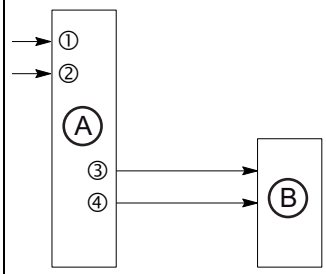
Any input that is not used for other applications can be used as the near home input.

Single pulse input driver (pulse and direction output method)

One output point is used as the pulse output and the other output is used as the direction output.

Set the control code for **F171** to pulse and direction.

Up to two driver systems can be connected.



Using channel		0	2
Ⓐ	PLC		
Ⓑ	Motor driver		
①	Home input	X2	X6
②	Near home input (see note)	e.g. X0	e.g. X1
③	Pulse output	Y0	Y4
④	Direction output	Y1	Y5

◆NOTE

Any input that is not used for other applications can be used as the near home input.

8.4.3 Instructions and System Variables

Use the following instructions to perform various positioning tasks:

Type of control	Instruction	Description
Target value match ON (pulse output)	F166_PulseOutput_Set	If the elapsed value matches the target value of the selected pulse output channel, the specified output immediately turns to TRUE.
Target value match OFF (pulse output)	F167_PulseOutput_Reset	If the elapsed value matches the target value of the pulse output channel, the specified output immediately turns to FALSE.
Trapezoidal control	F171_PulseOutput_Trapezoidal	This instruction automatically performs trapezoidal control according to the parameters in the specified DUT.
Home return	F177_PulseOutput_Home	This instruction performs a home return according to the parameters in the specified DUT.
JOG operation	F172_PulseOutput_Jog	This instruction is used for JOG operation.
JOG operation (input controlled)	F171_PulseOutput_Jog_Positioning	The specified number of pulses is output after the position control trigger input has turned to TRUE. A deceleration is performed before the target value is reached and pulse output stops.
Data table control	F174_PulseOutput_DataTable	This instruction performs rectangular control according to the parameters in the specified DUT with an arbitrary number of different speeds and target values.
Linear interpolation control	F175_PulseOutput_Linear	Pulses are output from two channels in accordance with the parameters in the specified DUT, so that the path to the target position forms a straight line.

Counter and pulse output settings as well as elapsed values are stored in special data registers. The pulse output status is stored in special internal relays. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering

a declaration in the POU header. Please refer to the FPWIN Pro online help for detailed information on using system variables.

Using the pulse output control flag

The relay is TRUE if a pulse output instruction is being executed. Use this flag to prohibit the simultaneous execution of other pulse output instructions on the specified channel, and to verify completion of the execution.

◆NOTE

The status of the high-speed counter control flag or pulse output control flag may change while a scan is being carried out. For example, if the flag is used more than once as an input condition, different statuses may exist within one scan. To ensure proper execution of the program, the status of the special internal relay should be copied to a variable at the beginning of the program.

Channel and pulse output numbers

Channel no.	Interpolation axis [1)]	Pulse output	Pulse output method	
			CW/CCW	Pulse/direction
0	x	Y0	CW	Pulse
		Y1	CCW	Direction
1	y	Y2	CW	Pulse
		Y3	CCW	Direction
2	x	Y4	CW	Pulse
		Y5	CCW	Direction
3	y	Y6	CW	Pulse
		Y7	CCW	Direction

1) For F175_PulseOutput_Linear

◆NOTE

For interpolation, channel 0 and 1 or channel 2 and 3 are used as pairs. You may only specify 0 or 2 (for C14T: 0 only).

System variables for memory areas used

Description		System variable	Address
Pulse output: control flag for channel	0	sys_bIsPulseChannel0Active	R9120
	1	sys_bIsPulseChannel1Active	R9121
	2	sys_bIsPulseChannel2Active	R9122
	3	sys_bIsPulseChannel3Active	R9123
Pulse output: elapsed value for channel	0	sys_diPulseChannel0ElapsedValue	DDT90400
	1	sys_diPulseChannel1ElapsedValu	DDT90410
	2	sys_diPulseChannel2ElapsedValue	DDT90420
	3	sys_diPulseChannel3ElapsedValue	DDT90430

Description		System variable	Address
Pulse output: target value for channel	0	sys_diPulseChannel0TargetValue	DDT90402
	1	sys_diPulseChannel1TargetValue	DDT90412
	2	sys_diPulseChannel2TargetValue	DDT90422
	3	sys_diPulseChannel3TargetValue	DDT90432
Corrected initial speed for channel[1)]	0	sys_iPulseChannel0CorrectedInitialSpeed	DT90406
	1	sys_iPulseChannel1CorrectedInitialSpeed	DT90416
	2	sys_iPulseChannel2CorrectedInitialSpeed	DT90426
	3	sys_iPulseChannel3CorrectedInitialSpeed	DT90436
Corrected final speed for channel[1)]	0	sys_iPulseChannel0CorrectedFinalSpeed	DT90407
	1	sys_iPulseChannel1CorrectedFinalSpeed	DT90417
	2	sys_iPulseChannel2CorrectedFinalSpeed	DT90427
	3	sys_iPulseChannel3CorrectedFinalSpeed	DT90437
Acceleration forbidden area starting position for channel[1)]	0	sys_diPulseChannel0AccelerationForbiddenAreaStarting Position	DDT90408
	1	sys_diPulseChannel1AccelerationForbiddenAreaStarting Position	DDT90418
	2	sys_diPulseChannel2AccelerationForbiddenAreaStarting Position	DDT90428
	3	sys_diPulseChannel3AccelerationForbiddenAreaStarting Position	DDT90438
Pulse output: control code monitor for channel	0	sys_wPulseChannel0ControlCode	DT90380
	1	sys_wPulseChannel1ControlCode	DT90381
	2	sys_wPulseChannel2ControlCode	DT90382
	3	sys_wPulseChannel3ControlCode	DT90383
High-speed counter or pulse output control code		sys_wHscOrPulseControlCode	DT90052

1) For F171_PulseOutput_Jog_Positioning, F171_PulseOutput_Trapezoidal, F172_PulseOutput_Jog

8.4.3.1 Writing the Pulse Output Control Code

The special data register where the high-speed counter and pulse output control code are stored can be accessed with the system variable sys_wHscOrPulseControlCode. (The system variable sys_wHscOrPulseControlCode corresponds to special data register DT90052.)

The control code settings for each channel can be monitored using the system variables sys_wHscChannelxControlCode or sys_wPulseChannelxControlCode (where x=channel number). The settings of this system variable remain unchanged until another setting operation is executed.

Operations performed by the pulse output control code:

- Setting/resetting near home input
- Continuing/stopping pulse output (forced stop)
- Enabling/disabling counting operations
- Resetting the elapsed value (software reset) of the high-speed counter

- Clearing high-speed counter and position control instructions (only)

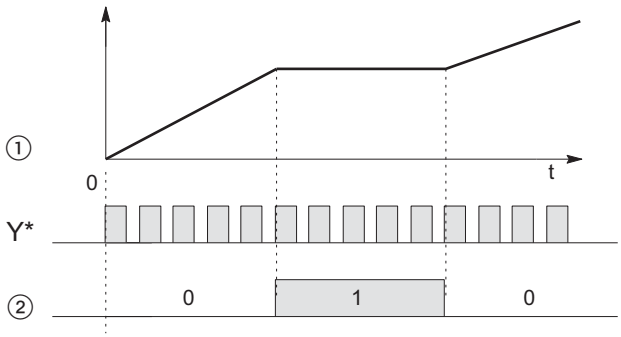
Setting/resetting near home input

The near home bit is retained. Set this bit to FALSE right after setting it to TRUE to be able to set the near home input a second time during a home return.

Continuing/stopping pulse output (forced stop)

By setting bit 3 of the data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE pulse output is stopped. The possibility of a forced stop should be provided for in every program using pulse output instructions. Reset bit 3 to FALSE to continue pulse output.

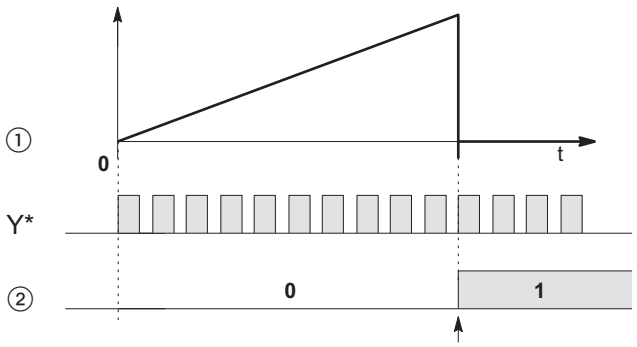
Enabling/disabling counting operations



Y*	Pulse output
①	Elapsed value
②	Bit 1 of pulse output control code (count)

When bit 1 of the control code is set to TRUE, counting is prohibited and the elapsed value keeps its current value. Counting is continued when bit 1 is reset to FALSE.

Resetting the elapsed value (software reset) of the high-speed counter to 0



Y*	Pulse output
①	Elapsed value
②	Bit 0 of pulse output control code (software reset)

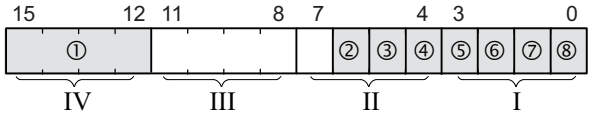
When bit 0 of the control code is set to TRUE, a software reset is performed and the elapsed value is set to 0. The elapsed value keeps the value 0 until bit 0 is reset to FALSE.

Clearing high-speed counter and position control instructions

To cancel execution of a pulse output instruction, set bit 2 of the data register storing the pulse output control code (sys_wHscOrPulseControlCode) to TRUE. The pulse output control flag will then change to FALSE. To reenable execution of the instruction, reset bit 2 to FALSE.

Control code settings

Bits 0–15 of the control code are allocated in groups of four. The bit setting in each group is represented by a hex number (e.g. 0002 0001 0000 1001 = 16#2109).



Group IV	①	Channel number (channel n: 16#n)	
Group III		1 (fixed)	
Group II	②	Position control start request	
		0: disabled	1: enabled
	③	Decelerated stop request	
		0: disabled	1: enabled
	④	Near home input (bit 4) (see note)	
		0: FALSE	1: TRUE
Group I	⑤	Pulse output (bit 3)	
		0: continue	1: stop
	⑥	Clear pulse output control (bit 2)	
		0: continue	1: stop
	⑦	Count (bit 1)	
		0: permit	1: prohibit
	⑧	Reset elapsed value to 0 (bit 0)	
		0: no	1: yes

Example: 16#2109

Group	Value	Description	
IV	2	Channel number: 2	
III	1	(fixed)	
II	0	Position control start request: disabled	
		Decelerated stop request: disabled	
		Near home input: FALSE	
I	9	Hex 9 corresponds to binary 1001	
		Pulse output: stop (bit 3)	1
		Clear pulse output control (bit 2)	0
		Count: permit (bit 1)	0
		Reset elapsed value to 0: yes (bit 0)	1

◆**NOTE**

- **Performing a forced stop may cause the elapsed value at the PLC output side to differ from the elapsed value at the motor input side. Therefore, you must execute a home return after pulse output has stopped.**
- **Setting the near home input is not possible if counting is prohibited or if a software reset is performed.**

◆REFERENCE

Please refer to the FPWIN Pro online help for programming examples.

8.4.3.2 Writing and Reading the Elapsed Value of the Pulse Output

The elapsed value is stored as a double word in the special data registers. Access the special data registers using the system variable sys_diHscChannelxElapsedValue (where x=channel number).

System variables for memory areas used:

Description		System variable	Address
Pulse output: elapsed value for channel	0	sys_diPulseChannel0ElapsedValue	DDT90400
	1	sys_diPulseChannel1ElapsedValue	DDT90410
	2	sys_diPulseChannel2ElapsedValue	DDT90420
	3	sys_diPulseChannel3ElapsedValue	DDT90430

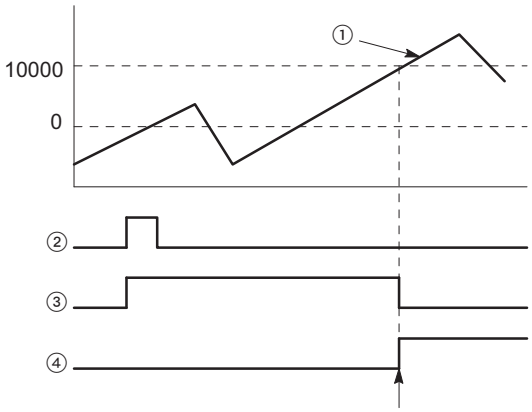
◆REFERENCE

Please refer to the FPWIN Pro online help for programming examples.

8.4.3.3 F166_PulseOutput_Set, Target Value Match ON

If the elapsed value matches the target value of the selected pulse output channel, the specified output immediately turns to TRUE.

Pulse output characteristics



10000	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	"Output control active" flag
④	PLC output

The PLC output turns to TRUE when the elapsed value matches the target value. In addition, the "Output control active" flag turns to FALSE and the instruction is deactivated.

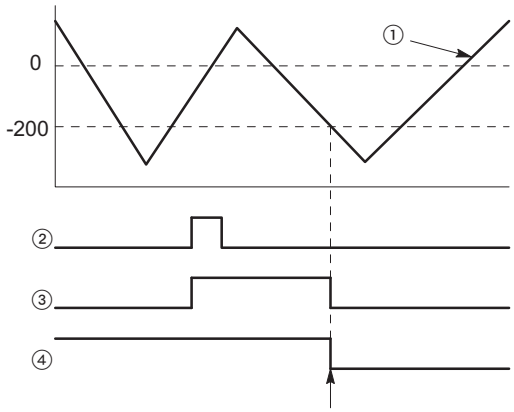
◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.4 F167_PulseOutput_Reset, Target Value Match OFF

If the elapsed value matches the target value of the pulse output channel, the specified output immediately turns to FALSE.

Pulse output characteristics



10000	Target value
①	Elapsed value of high-speed counter
②	Execution condition
③	"Output control active" flag
④	PLC output

The PLC output turns to FALSE when the elapsed value matches the target value. In addition, the "Output control active" flag turns to FALSE and the instruction is deactivated.

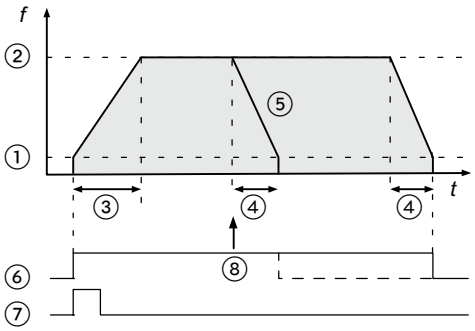
◆ REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.5 F171_PulseOutput_Trapezoidal, Trapezoidal Control

This instruction automatically performs trapezoidal control according to the parameters in the specified DUT. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Pulse output characteristics

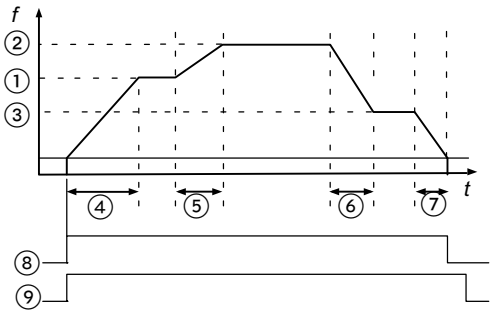


①	Initial and final speed	⑤	Target value
②	Target speed	⑥	Pulse output control flag
③	Acceleration time	⑦	Execution condition
④	Deceleration time	⑧	Decelerated stop request

Type 0: The difference between target speed and initial speed determines the slope of the acceleration ramp. The difference between target speed and final speed determines the slope of the deceleration ramp.

Type 1: The difference between the maximum speed of 50kHz and the final speed determines the slope of the deceleration ramp. The difference between the maximum speed of 50kHz and the initial speed determines the slope of the acceleration ramp.

Changing the target speed during pulse output



Type 1: The speed can be changed within the range of the maximum speed (50kHz).

①	Target speed	⑥	Deceleration
②	1st change of target speed	⑦	Deceleration time
③	2nd change of target speed	⑧	Pulse output control flag
④	Acceleration time	⑨	Execution condition
⑤	Acceleration		

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.6 F171_PulseOutput_Jog_Positioning, JOG Operation and Positioning

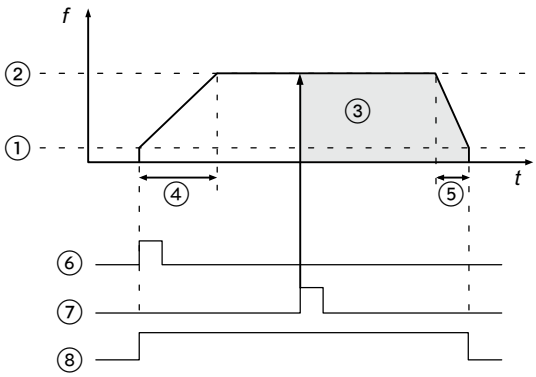
The specified number of pulses is output after the position control trigger input has turned to TRUE. A deceleration is performed before the target value is reached and pulse output stops. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Select one of two different operation modes:

Type 0: The speed can be changed within the range of the specified target speed.

Type 1: The target speed can be changed once when the position control trigger input turns to TRUE.

Pulse output characteristics

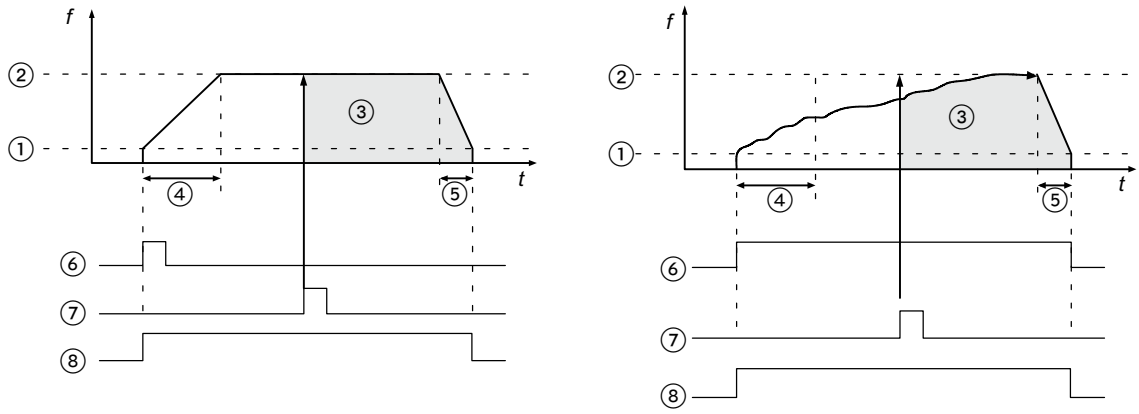


①	Initial and final speed	⑤	Deceleration time
②	Target speed	⑥	Execution condition
③	Target value	⑦	Position control trigger input
④	Acceleration time	⑧	Pulse output control flag

JOG Operation Type 0

The target speed can be changed during pulse output. The speed can be changed within the range of the specified target speed.

Without changing the target speed: With changing the target speed:

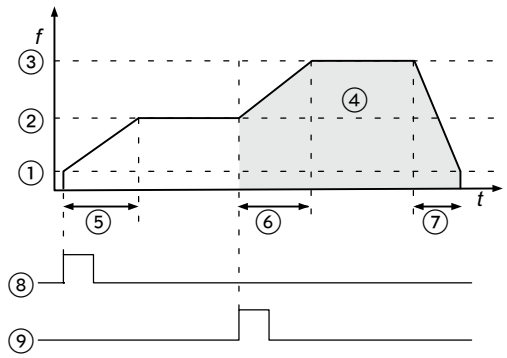


①	Initial and final speed	⑤	Deceleration time
②	Target speed	⑥	Execution condition
③	Target value	⑦	Position control trigger input
④	Acceleration time	⑧	Pulse output control flag

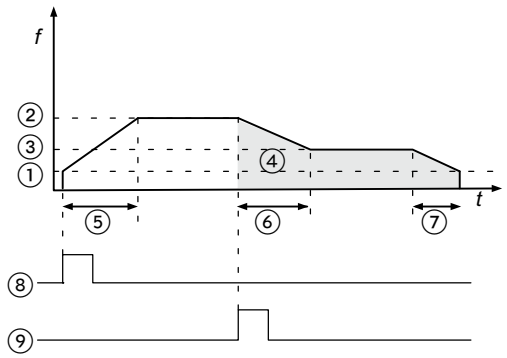
JOG Operation Type 1

The target speed can be changed once when the position control trigger input turns to TRUE.

Target speed 1 < target speed 2:



Target speed 1 > target speed 2:



①	Initial and final speed	⑥	Change time
②	Target speed 1	⑦	Deceleration time
③	Target speed 2	⑧	Execution condition
④	Target value	⑨	Position control trigger input
⑤	Acceleration time		

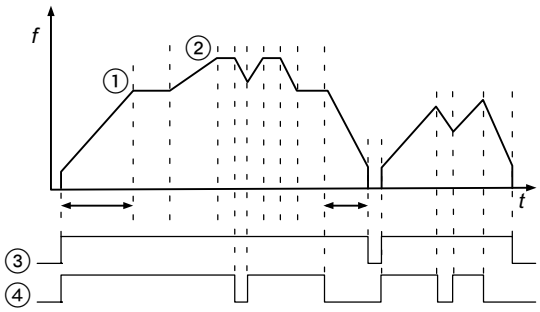
◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.7 F172_PulseOutput_Jog, JOG Operation

This instruction is used for JOG operation. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

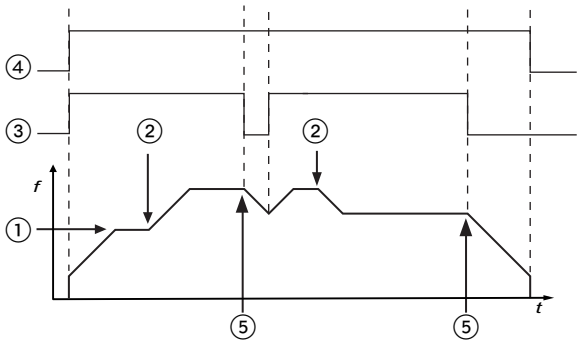
Pulse output characteristics



①	Target speed 1	③	Pulse output control flag
②	Target speed 2	④	Execution condition

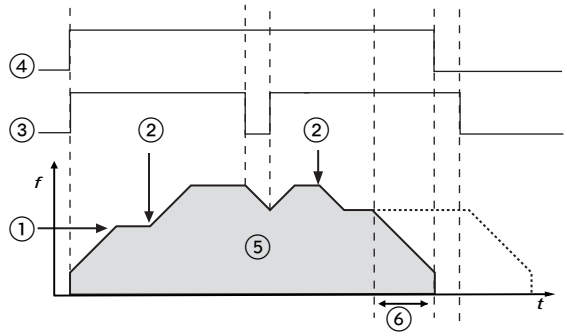
Select one of two different operation modes:

- Mode with no target value (type 0): Pulses are output in accordance with the conditions set in the DUT as long as the execution condition is TRUE. A decelerated stop begins whenever the execution condition is FALSE.



①	Initial and final speed	④	Pulse output control flag
②	Change of target speed	⑤	Decelerated stop
③	Execution condition		

- Target value match stop mode (type 1): Output stops when the target value is reached. Set this mode in the control code, and specify the target value (an absolute value) in the DUT. A decelerated stop is performed when the target value has been reached. Deceleration is performed within the specified deceleration time.



①	Initial and final speed	④	Pulse output control flag
②	Change of target speed	⑤	Target value
③	Execution condition	⑥	Deceleration time

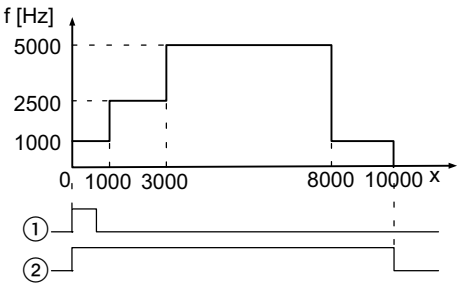
◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.8 F174_PulseOutput_DataTable, Data Table Control

This instruction performs rectangular control according to the parameters in the specified DUT with an arbitrary number of different speeds and target values. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Pulse output characteristics



x	Elapsed value of high-speed counter
①	Execution condition
②	Pulse output control flag

- Pulses are output at the specified frequency until the target value is reached. Then the frequency changes to the second frequency value and pulse output continues until the second target value is reached, and so forth.

- Pulse output stops when the last target value is reached.

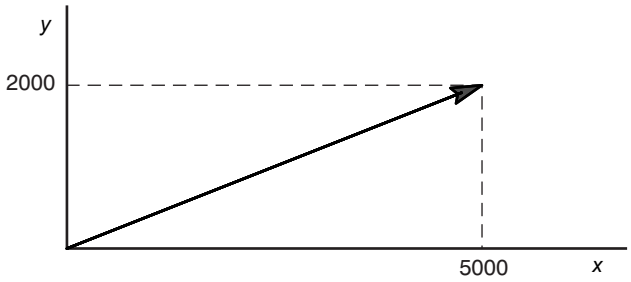
◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.9 F175_PulseOutput_Linear, Linear Interpolation

Pulses are output from two channels in accordance with the parameters in the specified DUT, so that the path to the target position forms a straight line. Pulses are output from the specified channel when the control flag for this channel is FALSE and the execution condition is TRUE.

Pulse output characteristics



5000	X-axis target value (channel 0)
2000	Y-axis target value (channel 1)

Pulses are output from the X-axis (channel 0) and the Y-axis (channel 1), so that the initial speed is 500Hz, the target speed is 5kHz, and the acceleration time and deceleration time is 300ms. The two axes are controlled so that a linear path is followed to the target position.

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.4.3.10 F177_PulseOutput_Home, Home Return

This instruction performs a home return according to the parameters in the specified DUT.

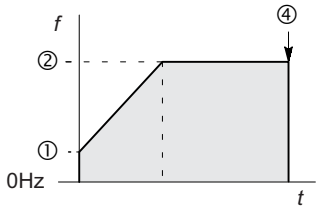
After a drive system has been switched on, there is a difference between the internal position value (elapsed value) and the mechanical position of the axis; this difference cannot be predetermined. The internal value must be synchronized with the actual position value of the axis. This is done by means of a home return, during which a position value is registered at a known reference point (home).

During execution of a home return instruction, pulses are continuously output until the home input is enabled. The I/O allocation is determined by the channel used.

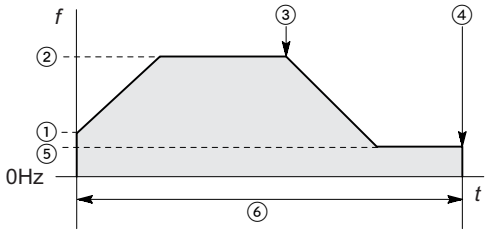
Select one of two different operation modes:

- Type 0: The home input is effective regardless of whether or not there is a near home input, whether deceleration is taking place, or whether deceleration has been completed.

Without near home input:

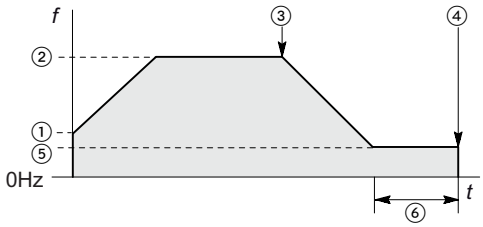


With near home input:



①	Initial speed	④	Home input: TRUE
②	Target speed	⑤	Creep speed
③	Near home input: TRUE	⑥	Home input is effective at any time.

- Type 1: The home input is effective only after deceleration (started by near home input) has been completed.



①	Initial speed	④	Home input: TRUE
②	Target speed	⑤	Creep speed
③	Near home input: TRUE	⑥	Home input is effective only after deceleration

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

8.5 PWM Output Function

8.5.1 PWM Output Function

Use the instruction F173_PulseOutput_PWM. This instruction delivers a pulse width modulated output signal according to the specified DUT.

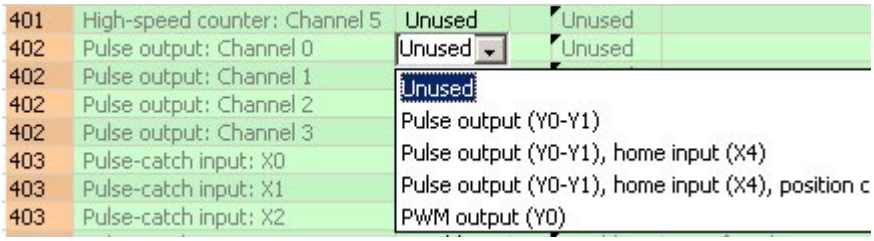
The PWM output status is stored in special internal relays. To access special data registers and special internal relays, use the PLC-independent system variables. You can insert system variables directly into the POU body: Use the "Variables" dialog without entering a declaration in the POU header. Please refer to the FPWIN Pro online help for detailed information on using system variables.

Setting system registers

When using the PWM output function, specify the desired PWM output in the system registers.

◆**Procedure**

1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **Double-click "High-Speed Counter, Pulse-Catch Input, Interrupt Input"**
4. **Specify the PWM output for the channel used**


401 High-speed counter: Channel 5 Unused Unused
402 Pulse output: Channel 0 Unused Unused
402 Pulse output: Channel 1
402 Pulse output: Channel 2
402 Pulse output: Channel 3
403 Pulse-catch input: X0
403 Pulse-catch input: X1
403 Pulse-catch input: X2
Unused
Pulse output (Y0-Y1)
Pulse output (Y0-Y1), home input (X4)
Pulse output (Y0-Y1), home input (X4), position c
PWM output (Y0)

◆REFERENCE

Please refer to the FPWIN Pro online help for details and a programming example.

Channel and pulse output numbers

Channel no.	Pulse output
0	Y0
1	Y2
2	Y4
3	Y6

System variables for memory areas used

Description		System variable	Address
Pulse output: control flag for channel	0	sys_bIsPulseChannel0Active	R9120
	1	sys_bIsPulseChannel1Active	R9121
	2	sys_bIsPulseChannel2Active	R9122
	3	sys_bIsPulseChannel3Active	R9123

Chapter 9

Security Functions

9.1 Security Function Types

The following security settings are available:

- PLC program upload protection
- password protection
- security settings for FP Memory Loader

9.2 Security Settings in FPWIN Pro

In FPWIN Pro, **Online** → **Security Settings** opens a dialog that displays the current security settings and enables you to protect your PLC.

The LEDs in the dialog display the PLC's current protection status. To display a tool tip, hold the cursor over the LED for approximately 2s.

◆REFERENCE

For a detailed description of the options in the Security Settings dialog box, please refer to the FPWIN Pro online help.

9.2.1 Upload Protection

When upload protection is enabled, you cannot:

- upload projects or program code to a PC
- upload system registers to a PC

Data can be lost permanently - even if you know the password!

When using this function, be sure to back up your programs! The program on your PLC will not be recoverable, either by a person knowing the password or by customer support.

You can cancel the setting for this function using FPWIN Pro. However, all programs, system registers and password information will be deleted!

If upload protection is enabled, you can edit files on the PLC while online using FPWIN Pro. However, programs will be corrupted if the program in FPWIN Pro and the program on the PLC are not identical.

◆**NOTE**

Even if upload protection has been set, upload to the FP Memory Loader is possible. If you are using FP Memory Loader version 2 or higher, you can enable or disable program upload to the FP Memory Loader or program transfer between two PLCs using the FP Memory Loader. For details, see "FP Memory Loader" on page 205.

9.2.2 PLC Protection (Password Protection)

You can set a new password with up to 8 characters, or change an existing one.

To access a PLC for which a password has been set, a login is required whenever the power is turned on.

To set a password, you can use:

- the programming tool
- the SYS1 instruction

- Be sure to memorize your password. Without the password, you will not be able to read programs on password-protected PLCs.
- If you have forgotten your password, our support team will not be able to reset it for you.
- If you are not logged in, [Clear Password] will erase not only the password but also the program and parameters stored in the PLC's comment memory.

◆REFERENCE

For details on the SYS1 instruction, please refer to your Programming Manual or to the FPWIN Pro online help.

9.3 FP Memory Loader

The FP Memory Loader V2.0 or higher (AFP8670/AFP8671) can be used to transfer a program from one PLC to another.

To prevent unauthorized copying of user programs, you must enable upload protection. This function is recommended for users who manage original programs on a PC.

In FPWIN Pro, **Online → Security Settings** opens the Security Settings dialog box, which offers two security settings for the FP Memory Loader:

- Upload protection
- Download protection

9.3.1 Upload Protection

Upload protection prevents programs from being uploaded to an FP Memory Loader.

◆Procedure

1. **Online → Security Settings**

   The Security Settings dialog box opens.

2. **Select "Enable upload protection"**

3. **Enter the password**

4. **Choose [Set Protection] or [Change Protection]**

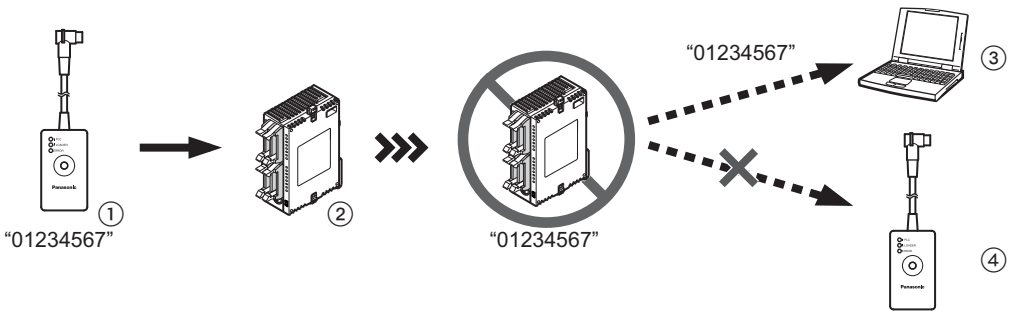
   When defining security settings for the first time, choose [Set Protection].

   To change existing security settings, choose [Change Protection].

5. **Download program from source PLC to FP Memory Loader**

6. **Transfer program to target PLC**

   After program download from the FP Memory Loader to the target PLC, this PLC is now upload-protected.



Program upload can be disabled in the Security Settings dialog box (see table below)

①	The FP Memory Loader contains a password- and upload-protected program: Password: 01234567 Upload protection: enabled
②	The security settings are transferred together with the program to the target PLC. The target PLC is now doubly protected:
③	Program upload to a PC requires password entry.
④	Upload to an FP Memory Loader is not possible, even if the source PLC and target PLC are protected by identical passwords ("01234567").

9.3.2 Download Protection

Using download protection, you can transfer a program from one PLC to another with the FP Memory Loader, provided the passwords the two PLCs' passwords are identical.

◆Procedure

1. **Online → Security Settings**

   The Security Settings dialog box opens.

2. **Select "Allow download to PLC only if password in the PLC is the same"**

3. **Enter the password**

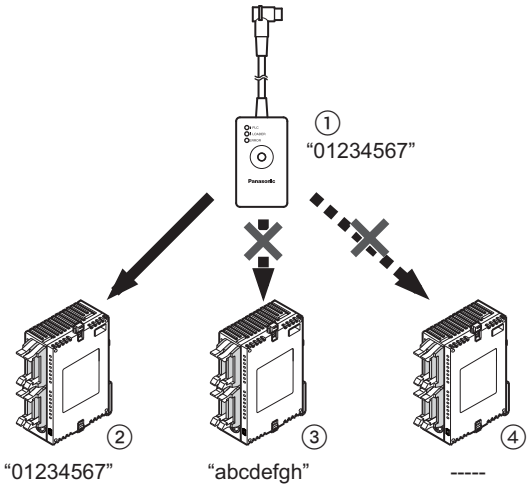
4. **Choose [Set Protection] or [Change Protection]**

   When defining security settings for the first time, choose [Set Protection].

   To change existing security settings, choose [Change Protection].

5. **Download program from source PLC to FP Memory Loader**

6. Transfer program to target PLC



Programs can only be downloaded to PLCs that are protected by identical passwords (see table below)

①	The FP Memory Loader contains a password protected program: Password: 01234567
②	Download is possible only if the target PLC is protected by the same password ("01234587").
③	Download to a target PLC that is protected by a different password ("abcdefgh") is not possible.
④	Download to a target PLC that is not password-protected (-----) is not possible.

◆CAUTION

During program download from the FP Memory Loader to a target PLC, the password set on the source PLC may be changed under certain conditions:

Security setting on FP Memory Loader	Password setting on target PLC after download
No password set	Password will be cleared.
8-digit password set, "Allow download to PLC only if password in the PLC is the same" setting disabled	Password will be overwritten with new 8-digit password.
8-digit password set, "Allow download to PLC only if password in the PLC is the same" setting enabled	Password will not be changed (no download possible).

Chapter 10

Other Functions

10.1 F-ROM Backup (P13_EPWT)

Data registers of 32765 words can be written to the built-in F-ROM of the FP0R CPU using the instruction P13_EPWT.

Writing can be performed up to 10000 times. After that, the correct operation cannot be guaranteed.

If the power supply turns off while the instruction **P13_EPWT** is being executed or during online editing, data in the hold area may be lost.

◆REFERENCE

For details, please refer to the FPWIN Pro Programming Manual or to the FPWIN Pro online help.

10.2 Sampling Trace

Using the sampling trace function, the current contact conditions and/or the variable values can be displayed on a time axis. After data recording in the PLC has been completed, the data is loaded into FPWIN Pro. Sampling parameters such as the sampling time and trigger conditions can be set in FPWIN Pro.

A maximum of 16 Boolean variables and three 16-bit variables can be traced per sampling.

◆REFERENCE

For details, please refer to the FPWIN Pro Programming Manual or to the FPWIN Pro online help.

10.3 Input Time Constants

You can specify input time constants in order to negate the effects of noise or bouncing, e.g. for a switching device.

To set the time constants, use the system registers or the instruction F182_FILTER.

Time constant settings are invalid if the input is used as a high-speed counter, pulse catch, or interrupt input.

◆REFERENCE

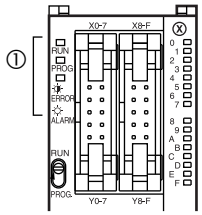
For details, please refer to the FPWIN Pro Programming Manual or to the FPWIN Pro online help.

Chapter 11

Troubleshooting

11.1 LED Display for Status Condition

When an error occurs, the status of the status indicator LEDs on the CPU changes as shown in the table below.



① Status indicator LEDs

Status indicator LEDs on CPU

	LED status			Description	Operation status
	RUN	PROG.	ERROR/ ALARM		
Normal condition	On	Off	Off	Normal operation	Continue
	Off	On	Off	PROG mode	Stop
	Flashes	Flashes	Off	Forcing on/off in RUN mode	Continue
Abnormal condition	On	Off	Flashes	A self-diagnostic error has occurred	Continue
	Off	On	Flashes	A self-diagnostic error has occurred	Stop
	Varies	Varies	On	System watchdog timer has been activated	Stop

11.2 Operation on Error

The CPU has a self-diagnostic function which identifies errors and stops operation if necessary. For some errors, the user may select whether operation shall continue or stop when the error occurs.

◆Procedure

1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **Double-click “Act on Error”**

   Select the desired setting for each type of error.

◆EXAMPLE

Operation is to continue even though a calculation error has occurred:

Set system register no. 26 "Operation error" to "Continue". Operation errors will be handled as an error, but operation will continue.

11.3 ERROR/ALARM LED is Flashing

Chec the error code using the programming tool.

◆Procedure

1. **In online mode: Monitor → PLC Status or**

   The error code is displayed in the "Self-diagnostic Error" section.

For error codes 20 or higher: a self-diagnostic error other than a syntax error has occurred.

There are three ways to clear the error:

- Choose [Clear] in the PLC Status dialog box while in PROG mode
- Turn the power supply off/on while in PROG mode (this clears all of the contents of the operation memory except hold type data)
- Execute the self-diagnostic error set instruction F148_ERR

◆NOTE

- **If the mode selector switch has been set to RUN, the error is cleared and at the same time operation is enabled. However, the error continues to be displayed unless the cause of the error has been eliminated.**
- **When an operation error (error code 45) occurs, the address at which the error occurred is stored in special data registers DT90017 (sys_iOperationErrorStepHold) and DT90018 (sys_iOperationErrorNonHold). If this happens, monitor the address at which the error occurred before cancelling the error.**

11.4 ERROR/ALARM LED is ON

If the ERROR/ALARM LED is on, the system watchdog timer has been activated and the operation of the PLC has been stopped. There are two ways to remedy the problem:

- Set the mode selector of the PLC from RUN to PROG mode and turn the power off and then on.
  - If the ERROR/ALARM LED turns on again, there is probably an abnormality in the CPU. Please contact your dealer.
  - Check if the ERROR/ALARM LED is flashing. See "ERROR/ALARM LED is Flashing" on page 216.
- Set the mode selector from PROG to RUN mode. If the ERROR/ALARM LED turns on, the program execution time is too long.
  - Check if instructions such as JP or LOOP are programmed in such a way that a scan can never finish.
  - Make sure that interrupt instructions are executed in succession.

11.5 All LEDs are OFF

If all LEDs are OFF, try the following:

- Check the power supply wiring.
- Check if the power supplied to the CPU is in the range of the rating. Be sure to check the fluctuation in the power supply.
- Disconnect the power supply wiring to the other devices if the power supplied to the CPU is shared with them.
  - If the LEDs on the CPU turn on at this moment, increase the capacity of the power supply or prepare another power supply for other devices.
  - Please contact your dealer for further information.

11.6 Diagnosing Output Malfunction

If the outputs do not function correctly, check the output side first and then the input side.

If the output indicator LEDs are ON:

- Check the wiring of the loads.
- Check if the power is properly supplied to the loads.
  - If the power is properly supplied to the load, there is probably an abnormality in the load. Check the load again.
  - If the power is not supplied to the load, there is probably an abnormality with the outputs. Please contact your dealer.

If the output indicator LEDs are OFF:

- Monitor the output condition using FPWIN Pro.
  - If the output monitored is TRUE, there is probably a duplicate output error.
- Set the output to TRUE by force using FPWIN Pro.
  - If the output indicator LED is turned ON, you must check the input side.
  - If the output indicator LED remains OFF, there is probably an abnormality with the outputs. Please contact your dealer.

If the input indicator LEDs are OFF:

- Check the wiring of the input devices.
- Check that the power is properly supplied to the input terminals.
  - If the power is properly supplied to the input terminal, there is probably an abnormality with the inputs. Please contact your dealer.
  - If the power is not supplied to the input terminal, there is probably an abnormality in the input device or input power supply. Check the input device and input power supply.

If the input indicator LEDs are ON:

Monitor the input condition using FPWIN Pro.

- If the input monitored is FALSE, there is probably an abnormality with the inputs. Please contact your dealer.
- If the input monitored is TRUE, check the leakage current at the input devices (e.g. two-wire type sensor) and check the program again, referring to the following:
  - Check for duplicate output errors and for outputs having been rewritten by high-level instructions.
  - Check the program flow when instructions such as MC or JP are used.

11.7 Password Protection Error Message

If a protect error message appears, a password has been set.

To access a PLC for which a password has been set, a login is required whenever the power is turned on.

◆Procedure

1. **Online → Security Settings**
2. **Enter your password under "PLC Access"**
3. **Choose [Login]**

11.8 PROG Mode does not Change to RUN

If PROG mode does not change to RUN, a syntax error or a self-diagnostic error that caused operation to stop has occurred.

- Check to see if the ERROR/ALARM LED is flashing. See "ERROR/ALARM LED is Flashing" on page 216.
- Locate the syntax error by executing **Monitor** → **PLC Status**

Chapter 12

Appendix

12.1 Specifications

12.1.1 General Specifications

Item		Description		
Rated operating voltage		24V DC		
Operating voltage range		20.4–28.8V DC		
Momentary power off time	**C10 C14 C16**	5ms at 20.4V, 10ms at 21.6V		
	C32 T32 F32	10ms at 20.4V		
Fuse		Built-in (cannot be replaced)		
Ambient temperature		0–+55°C		
Storage temperature		-40–+70°C (T32: -20–+70°C)		
Ambient humidity		10%–95% RH (at 25°C non-condensing)		
Storage humidity		10%–95% RH (at 25°C non-condensing)		
Breakdown voltage (cutoff current: 5mA)			Transistor types	Relay types
		Input terminals « Output terminals	500V AC for 1min	1500V AC for 1min
		Output terminals « Output terminals (of different COM terminals)	–	1500V AC for 1min
		Input terminals « Power supply terminal/Function earth	500V AC for 1min	500V AC for 1min
		Output terminals « Power supply terminal/Function earth	500V AC for 1min	1500V AC for 1min
		Function earth « Power supply terminal	500V AC for 1min	500V AC for 1min
Insulation resistance (measured with a 500V DC megger)		Input terminals « Output terminals	Min. 100Ω	Min. 100Ω
		Output terminals « Output terminals (of different COM terminals)	–	Min. 100Ω
		Input terminals « Power supply terminal/Function earth	Min. 100Ω	Min. 100Ω
		Output terminals « Power supply terminal/Function earth	Min. 100Ω	Min. 100Ω
		Function earth « Power supply terminal	Min. 100Ω	Min. 100Ω
Vibration resistance		5–9Hz, 1 cycle/min: single amplitude of 3.5mm 9–150Hz, 1 cycle/min: constant acceleration of 9.3m/s^2, 10min on 3 axes (in X, Y, and Z direction)		
Shock resistance		147m/s^2, 4 times on 3 axes (in X, Y, and Z direction)		
Noise immunity (Power supply terminal)		1000Vp-p, with pulse widths 50ns and 1μs (based on in-house measurements)		
Operation conditions		Free from corrosive gases and excessive dust		
Overvoltage category		II		
Pollution level		2		
Weight		C10: 100g, C14: 105g, C16: 85g, C32: 115g, T32: 115g, F32: 120g		

12.1.2 Performance Specifications

Item		C10, C14, C16	C32, T32, F32
Programming method/Control method		Relay symbol/cyclic operation	
Program memory	**Built-in memory**	F-ROM	
	Program capacity (steps)	16000	32000
	Online edit mode	Available (entire program)	
	Security function	Password protection (4-digit, 8-digit), upload protection	
Comment memory	**Memory capacity**	328kbyte	
	Online edit mode	Available (project information)	
I/O refresh time		≤0.2ms With expansion units: ≤0.2ms + (1 × no. of expansion units)ms	
Operation speed	**≤3000 steps**	Basic instructions: 0.08μs, timer instruction: 2.2μs High-level instructions: 0.32μs (MV instruction)	
	>3000 steps	Basic instructions: 0.58μs, timer instruction: 3.66μs High-level instructions: 1.62μs (MV instruction)	
Basic instructions		Approx. 110	
High-level instructions		Approx. 210	
Operation memory: Relays	**External input relays (X)**	1760[1)]	
	External output relays (Y)	1760[1)]	
	Internal relays (R)	4096	
	Special internal relays (R)	224	
	Timer relays/Counter relays (T/C)	1024[2)] Factory setting timers: 1008 points (T0–T1007) Factory setting counters: 16 points (C1008–C1023) Timer: 1–32767 (in units of 1ms, 10ms, 100ms, or 1s). Counter: 1–32767	
	Link relays (L)	2048	
Operation memory: Memory areas	**Data registers (DT)**	12315 words	32765 words
	Special data registers (DT)	440 words (DT90000–DT90443)	
	Link registers (LD)	256 words[1)]	
	Index registers (I)	14 words (I0–ID)	
Differential points		Depending on program capacity	
Master control relay points (MCR)		256	
Number of labels (JP and LOOP)		256	
Number of SFC steps		1000	
Number of subroutines		500	
Sampling trace		300 samples	1000 samples
		Per scan or per time interval Max. 16 Boolean variables and 3 16-bit variables per sampling	

Item		C10, C14, C16	C32, T32, F32
High-speed counter [1]		1-phase: 6 channels (max. 50kHz) 2-phase: 3 channels (max. 15kHz)	
Pulse output (not available for C10, C14) [1)2)]		4 channels (max. 50kHz)	
PWM output (not available for C10, C14) [1)2)]		4 channels (max. 4.8kHz)	
Pulse catch inputs		8 (including high-speed counter and interrupt input)	
Number of interrupt programs		8 external inputs (C10: 6) 1 periodical interrupt 4 target value match interrupts	
Periodical interrupt		0.5ms–1.5s (unit: 0.5ms), 10ms–30s (unit: 10ms)	
Constant scan		0.5ms–600ms (unit: 0.5ms)	
F-ROM backup [3]	**Using instructions F12 and P13**	All areas (32765 words)	
	Automatically when power is cut off	Counter relays: 16 (C1008–C1023) Internal relays: 128 (R2480–R255F) Data registers: 315 words	
		DT12000–DT12314	DT32450–DT32764
RAM backup (T32 and F32 only) [4]		T32: All areas (built-in backup battery)[5] F32: All areas	
Clock/calendar function [6]		Available for T32 only.	
Communication ports		TOOL port, USB port, RS232C COM port (product numbers AFPX10CX, AFPX14CX, AFPX16CX, and AFPX32CX only)	
Self-diagnostic function		E. g. watchdog timer, program syntax check (Watchdog timer: approx. 690ms)	

1) These are the specifications when the rated input voltage is 24V DC at 25°C. The frequency will decrease depending on voltage, temperature or usage condition.

   A total of 4 channels is available for pulse output and PWM output.

2) Pulse output can be specified up to 50kHz. PWM output can be specified up to 4.8kHz. A maximum error on the pulse width of 40μs may occur for the setting value depending on voltage, temperature or operating condition.

3) Writing is possible up to 10000 times.

4) All memory areas including timers/counters, internal relays, link relays, link registers and data registers can be backed up. Areas to be held and not to be held can be specified in the system registers.

5) The optional battery has not been charged when it is shipped from the factory. It must be charged before you can use it. The battery does not have an alarm function when it is running low. When it is empty, the hold areas will become indefinite. However, the values will be cleared to 0 the next time the power is turned on. (We recommend adding a program for clearing the data to 0 when the values in the hold areas become indefinite.)

6) Precision of calendar timer:
   - At 0°C: less than 104 seconds error per month.
   - At 25°C: less than 51 seconds error per month.
   - At 55°C: less than 155 seconds error per month

12.1.3 Communication Specifications

TOOL port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Slave Modem initialization Program controlled (in RUN mode only)

USB port

Item	Description
Standard (baud rate)	USB2.0 Fullspeed
Communication mode	MEWTOCOL-COM Slave

COM port

Item	Description
Interface	RS232C
Transmission distance	15m
Baud rate	2400, 4800, 9600, 19200, 38400, 57600, 115200bit/s
Communication method	Half-duplex
Synchronous method	Start stop synchronous system
Communication format	Data length: 7 bits/8 bits Parity: None/Odd/Even Stop bit: 1 bit/2 bits End code: CR/CR+LF/None/ETX Start code: No STX/STX
Data transmission order	Transmits from bit 0 character by character.
Communication mode	MEWTOCOL-COM Master/Slave Modem initialization Program controlled Modbus RTU Master/Slave PLC Link

Default settings

Port	Baud rate	Data length	Parity	Stop bit
TOOL port	9600bit/s	8 bits	Odd	1 bit
COM port	9600bit/s	8 bits	Odd	1 bit

12.1.4 Current Consumption

Type of unit		CPU [1]	Expansion unit [2]	Input circuit [3]	Output circuit [4]
FP0R CPU	FP0R-C10	≤100mA	–	≤15.9mA	–
	FP0R-C14	≤120mA	–	≤21.1mA	–
	FP0R-C16	≤70mA	–	≤21.1mA	≤20mA
	FP0R-C32 FP0R-T32 FP0R-F32	≤90mA	–	≤42.2mA	≤40mA
FP0 I/O expansion unit	FP0-E8X	≤10mA	—	≤34.4mA	–
	FP0-E8R	≤15mA	≤50mA	≤17.2mA	–
	FP0-E8YR	≤10mA	≤100mA	–	–
	FP0-E8YT/P	≤15mA	–	–	≤24mA
	FP0-E16X	≤20mA	–	≤68.8mA	–
	FP0-E16R		≤100mA	≤34.4mA	–
	FP0-E16T/P	≤25mA	–		≤24mA
	FP0-E16YT/P		–	–	≤48mA
	FP0-E32T/P	≤40mA	–	≤68.8mA	
	FP0-E32RS		≤200mA	≤69mA	–
FP0 analog unit	FP0-A04V	≤20mA	≤100mA	–	–
	FP0-A04I		≤130mA	–	–
	FP0-A21	≤20mA	≤100mA	–	–
	FP0-A80		≤60mA	–	–
	FP0-TC4/TC8/RTD6	≤25mA	–	–	–
FP0 intelligent unit	FP0-IOL	≤30mA	≤40mA	–	–
	FP0-CCLS	≤40mA		–	–
	FP0-DPS2	≤10mA	–	–	–
Communication cassette	FPG-COM1 FPG-COM2	≤20mA	–	–	–
	FPG-COM3 FPG-COM4	≤25mA	–	–	–
GT series programmable display (5V type)	AIGT0030B1 AIGT0030H1 AIGT0230B1 AIGT0230H1	≤80mA	–	–	–
C-NET adapter S2	AFP15402	≤50mA	–	–	–

1) The current consumed by the CPU power supply connector. If expansion units or intelligent units are added, the current is increased by the value indicated in the table.

2) The current consumed by the expansion unit power supply connector. If a unit is not listed in the table, it means that it has no power supply connector.

3) The current consumed by the input circuits of the various units. The value indicates the current that flows into the input circuit.

4) The current consumed by the output circuits of the various units. The value indicates the current used to drive the output circuits. The value does not include the load current value.

12.1.5 I/O Allocation

FP0R CPUs

CPU type		Number of I/O points	I/O addresses
C10	**Input**	6	X0–X5
	Output	4	Y0–Y3
C14	**Input**	8	X0–X7
	Output	6	Y0–Y5
C16	**Input**	8	X0–X7
	Output	8	Y0–Y7
C32/T32/F32	**Input**	16	X0–XF
	Output	16	Y0–YF

FP0 expansion units

I/O allocation is performed automatically when an expansion unit is added and is determined by the installation location.

Type of unit			Number of I/O points	Channel	Unit number (installation location)		
					1	2	3
FP0 I/O expansion unit							
	FP0-E8X	Input	8	–	X20–X27	X40–X47	X60–X67
	FP0-E8R	Input	4	–	X20–X23	X40–X43	X60–X63
		Output	4	–	Y20–Y23	Y40–Y43	Y60–Y63
	FP0-E8YR, E8YT, E8YP	Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
	FP0-E16X	Input	16	–	X20–X2F	X40–X4F	X60–X6F
	FP0-E16R, E16T, E16P	Input	8	–	X20–X27	X40–X47	X60–X67
		Output	8	–	Y20–Y27	Y40–Y47	Y60–Y67
	FP0-E16YT, E16YP	Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
	FP0-E32T, E32P, E32RS	Input	16	–	X20–X2F	X40–X4F	X60–X6F
		Output	16	–	Y20–Y2F	Y40–Y4F	Y60–Y6F
FP0 analog I/O unit FP0-A21		Input	16	0	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Input	16	1	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
		Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 A/D conversion unit FP0-A80 and FP0 thermocouple unit FP0-TC4, FP0-TC8		Input	16	0, 2, 4, 6	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Input	16	1, 3, 5, 7	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
FP0 D/A conversion unit FP0-A04V, FP0-A04I		Input	16	–	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
		Output	16	0, 2	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
		Output	16	1, 3	WY3 (Y30–Y3F)	WY5 (Y50–Y5F)	WY7 (Y70–Y7F)

Type of unit		Number of I/O points	Channel	Unit number (installation location)		
				1	2	3
FP0 RTD unit FP0-RTD6	Input	16	0, 2, 4	WX2 (X20–X2F)	WX4 (X40–X4F)	WX6 (X60–X6F)
	Input	16	1, 3, 5	WX3 (X30–X3F)	WX5 (X50–X5F)	WX7 (X70–X7F)
	Output	16	–	WY2 (Y20–Y2F)	WY4 (Y40–Y4F)	WY6 (Y60–Y6F)
FP0 I/O link unit FP0-IOL	Input	32	–	X20–X3F	X40–X5F	X60–X7F
	Output	32	–	Y20–Y3F	Y40–Y5F	Y60–Y7F

◆**NOTE**

- **The data for each channel of the A/D and D/A conversion units FP0-A80, FP0-TC4/TC8, FP0-A04V/I, and FP0-RTD6 is converted and loaded with a user program that includes a switching flag to convert the data in 16-bit words (see corresponding manuals).**

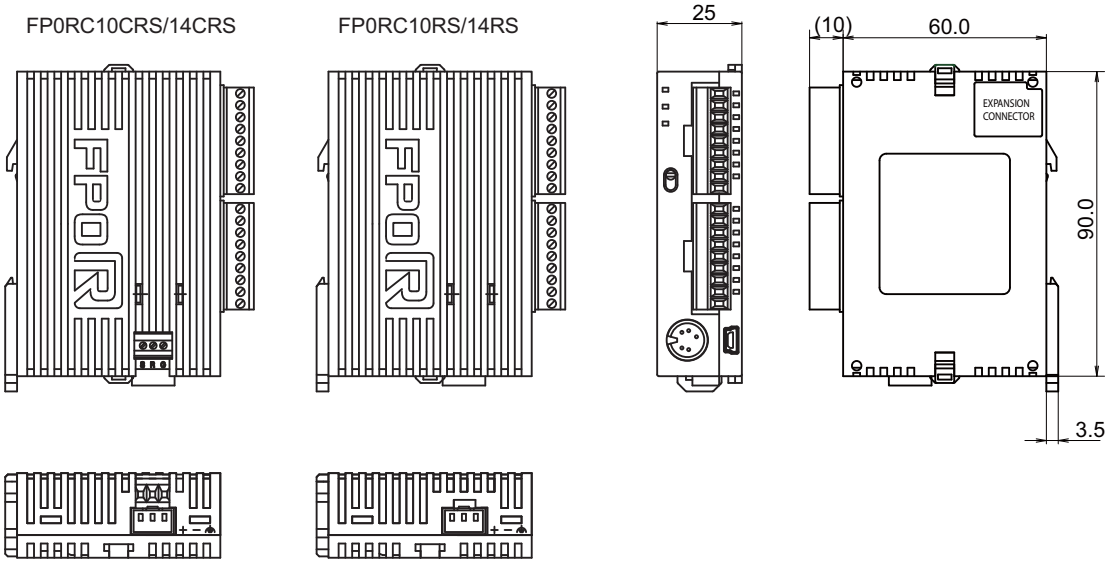
12.2 Dimensions

12.2.1 C10/C14 CPU (Terminal Block)

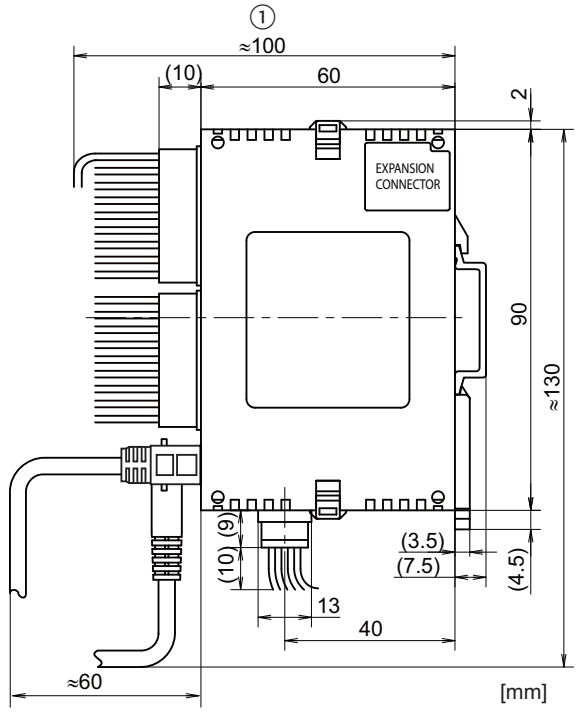
FP0RC10CRS/14CRS, FP0RC10RS/14RS

The same dimensions apply to the following FP0 expansion units:

- FP0-E8RS
- FP0-E16RS.



When mounting terminal block and power supply cable



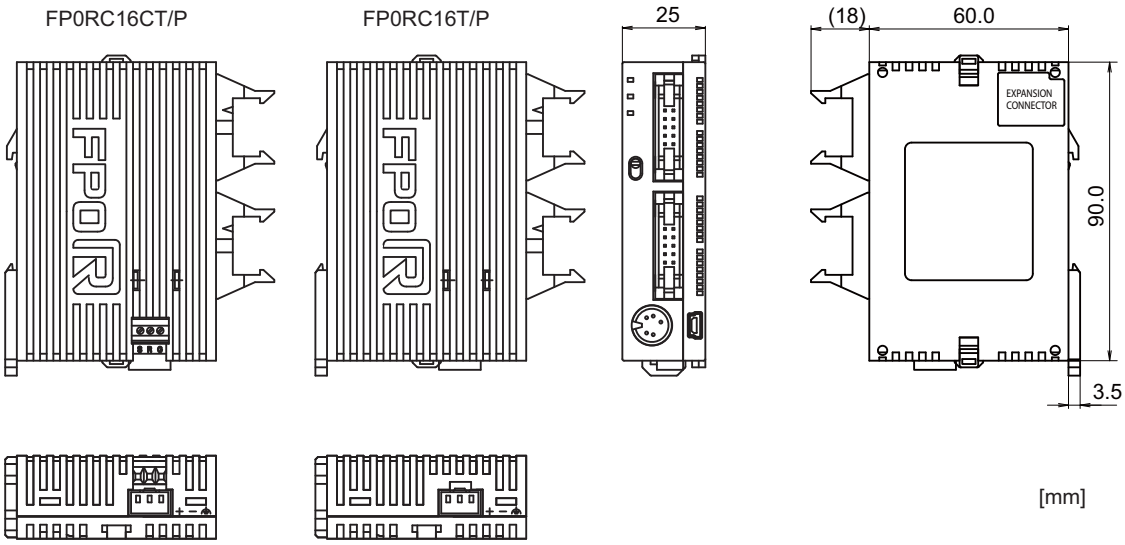
①	Maximum installation dimension

12.2.2 C16 CPU (MIL Connector)

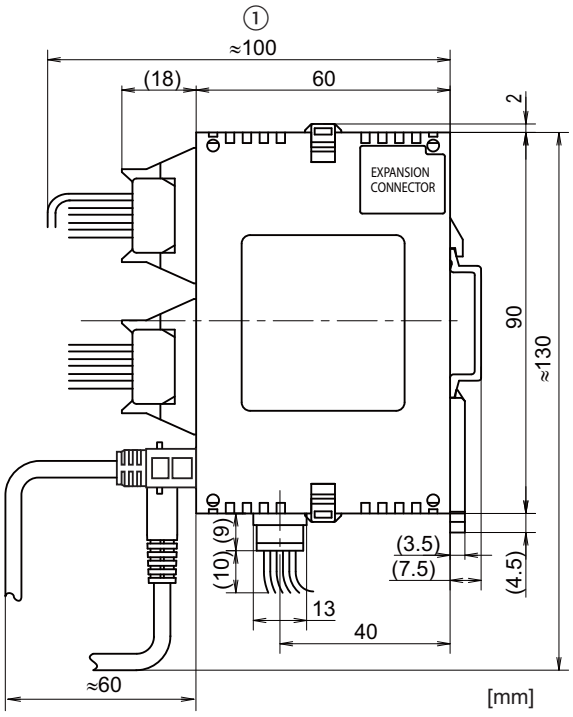
FP0RC16CT/P, FP0RC16T/P

The same dimensions apply to the following FP0 expansion units:

- FP0-E32T, FP0-E32P
- FP0-E16X, FP0-E16YT, FP0-E16YP, FP0-E16T, FP0-E16P
- FP0-E8X, FP0-E8YT, FP0-E8YP



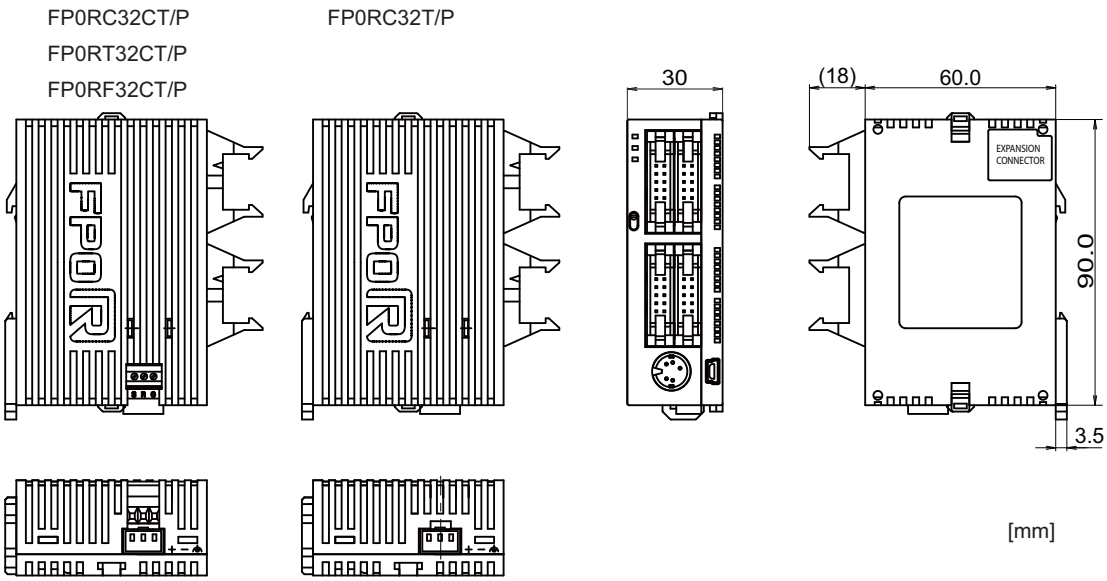
When mounting MIL connector and power supply cable



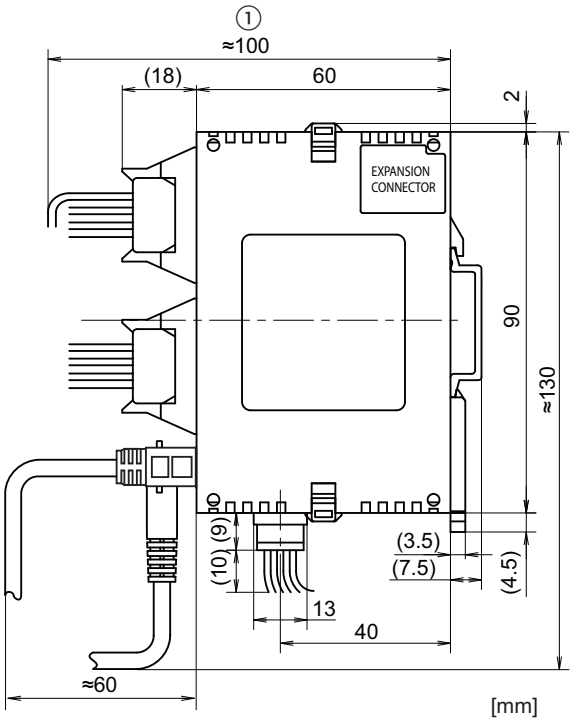
①	Maximum installation dimension

12.2.3 C32 CPU (MIL Connector)

FP0RC32CT/P, FP0RT32CT/P, FP0RF32CT/P, FP0RT32T/P



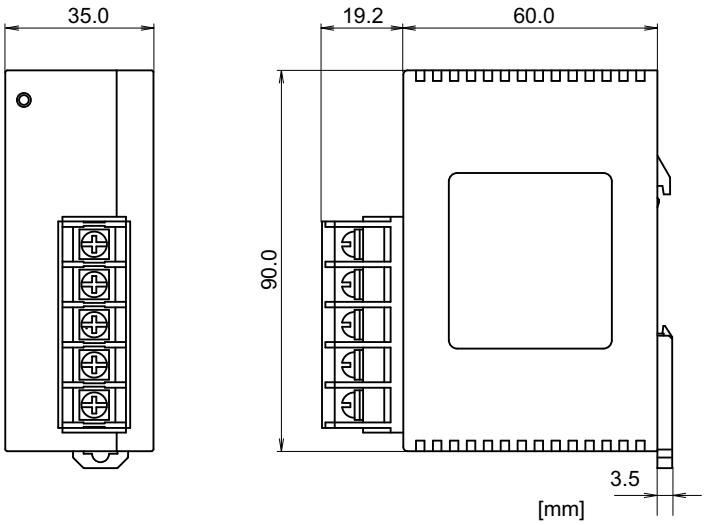
When mounting MIL connector and power supply cable



①	Maximum installation dimension

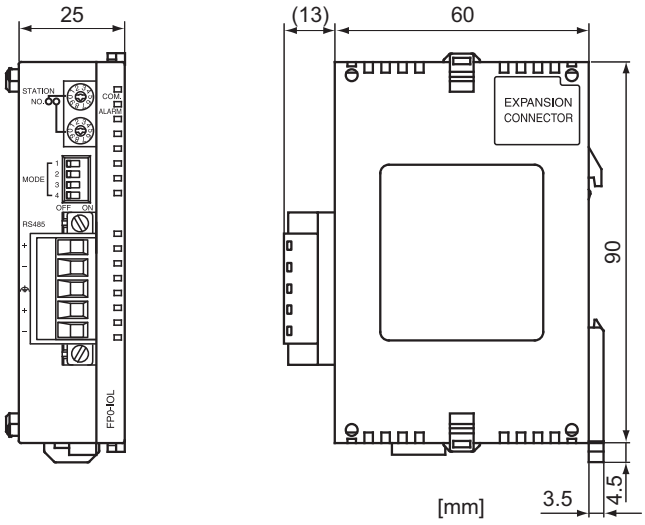
12.2.4 Power Supply Unit

FP0-PSA4

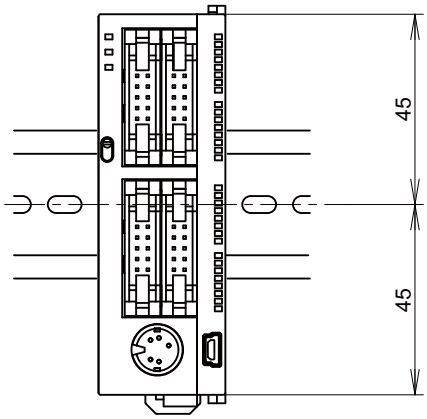


12.2.5 I/O Link Unit

AFP0732



12.2.6 Using DIN Rails



12.3 Relays and Memory Areas for FP0R

Relays [bits]

Type	Memory size	Available address area		Function
		FP	IEC	
External input relays [1]	1760	X0–X109F	%IX0.0– %IX109.15	Turn on or off based on external input.
External output relays [1]	1760	Y0–Y109F	%QX0.0– %QX109.15	Turn on or off external outputs based on the operation result.
Internal relays [2]	4096	R0–R255F	%MX0.0.0– %MX0.255.15	Used internally by the PLC program to store bit information.
Link relays [2]	2048	L0–L127F	%MX7.0.0– %MX7.127.15	Shared by multiple PLCs connected using PLC link.
Timer relays [2] [3]	1024	T0–T1007/ C1008-C1023	%MX1.0– %MX1.1007/ %MX2.1008– %MX2.1023	Turn on when the value set with a TM instruction for the timer with the same number has reached 0.
Counter relays [2] [3]	1024	C1008–C1023/ T0–T1007	%MX2.1008– %MX2.1023/ %MX1.0– %MX1.1007	Turn on when the value set with a CT instruction for the counter with the same number has reached 0.
Special internal relays	224	R9000–R913F	%MX0.900.0– %MX0.913.15	Turn on or off based on specific conditions. Used internally as a flag.

Memory area [words]

Type		Memory size	Available address area		Function
			FP	IEC	
External input relays [1]		110	WX0–WX109	%IW0– %IW109	Code for specifying 16 external input points as one word (16 bits) of data.
External output relays [1]		110	WY0–WY109	%QW0– %QW109	Code for specifying 16 external output points as one word (16 bits) of data.
Internal relays [2]		256	WR0–WR255	%MW0.0– %MW0.255	Code for specifying 16 internal relays as one word (16 bits) of data.
Link relays		128	WL0–WL127	%MW7.0– %MW7.127	Code for specifying 16 link relays as one word (16 bits) of data.
Data registers [2]	**C10, C14, C16**	12315	DT0–DT12312	%MW5.0– %MW5.12312	Data memory used in a program. Data is handled in 16-bit units (one word).
	C32, T32, F32	32763	DT0–DT32762	%MW5.0– %MW5.32762	
Link registers [2]		256	LD0–LD255	%MW8.0– %MW8.255	Data memory shared by multiple PLCs connected using PLC link. Data is handled in 16-bit units (one word).
Timer/counter set value area [2]		1024	SV0–SV1023	%MW3.0– %MW3.1023	Data memory for storing the set values of timers or counters. The values are stored by timer/counter number.

Type	Memory size	Available address area		Function
		FP	IEC	
Timer/counter elapsed value area [2]	1024	EV0–EV1023	%MW4.0– %MW4.1023	Data memory for storing the elapsed values during operation of timers or counters. The values are stored by timer/counter number.
Special data registers	440	DT90000– DT90439	%MW5.90000– %MW5.90439	Data memory for storing settings and error codes.

Memory area [double words]

Type		Memory size	Available address area		Function
			FP	IEC	
External input relays [1]		55	DWX0–DWX108	%ID0– %ID108	Code for specifying 32 external input points as a double word (32 bits) of data.
External output relays [1]		55	DWY0–DWY108	%QD0– %QD108	Code for specifying 32 external output points as a double word (32 bits) of data.
Internal relays [2]		128	DWR0–DWR254	%MD0.0– %MD0.254	Code for specifying 32 internal relay points as a double word (32 bits) of data.
Link relays		64	DWL0–DWL126	%MD7.0– %MD7.126	Code for specifying 32 link relay points as a double word (32 bits) of data.
Data registers [2]	**C10, C14, C16**	6157	DDT0– DDT12311	%MD5.0– %MD5.12311	Data memory used in a program. Data is handled in 32-bit units (double word).
	C32, T32, F32	16382	DDT0– DDT32761	%MD5.0– %MD5.32761	
Link registers [2]		128	DLD0–DLD126	%MD8.0– %MD8.126	Data memory shared by multiple PLCs connected using PLC link. Data is handled in 32-bit units (double word).
Timer/counter set value area [2]		512	DSV0–DSV1022	%MD3.0– %MD3.1022	Data memory for storing the set values of timers or counters. The values are stored by timer/counter number.
Timer/counter elapsed value area [2]		512	DEV0–DEV1022	%MD4.0– %MD4.1022	Data memory for storing the elapsed values during operation of timers or counters. The values are stored by timer/counter number.
Special data registers		220	DDT90000– DDT90438	%MD5.90000– %MD5.90438	Data memory for storing settings and error codes.

1) The number of points noted above is the number reserved as the calculation memory. The actual number of points available for use is determined by the hardware configuration.

2) There are hold and non-hold type memory areas. When the power supply turns off or the mode is changed from RUN to PROG mode, hold type areas are stored and non-hold type areas are reset.

C10/C14/C16/C32:The hold type and non-hold type areas are fixed. For information on the size of each area, refer to the performance specifications.

T32/F32: The settings of the hold type areas and non-hold type areas can be changed using the system registers.

T32: f the battery is empty and additional hold areas have been defined, the hold/non-hold operation becomes unstable. The data value will become indefinite. It is cleared to 0 the next time the power is turned on. See

"Backup and Clock/Calendar Functions" on page 31.

3) The number of points for timer and counter relays can be changed using system register 5. The numbers in the table are the default settings.

12.4 System Registers

System registers are used to set values (parameters) which determine operation ranges and functions used. Set values based on the use and specifications of your program. There is no need to set system registers for functions which will not be used.

12.4.1 Precautions When Setting System Registers

System register settings are effective from the time they are set.

However, MEWNET-W0 PLC link settings, input settings, TOOL and COM port communication settings become effective when the mode is changed from PROG to RUN. With regard to the modem connection setting, when the power is turned off and on or when the mode is changed from PROG to RUN, the PLC sends a command to the modem which enables it for reception.

After an initialization with **Online → Clear Program and Reset System Register**, all system register values (parameters) set will be reset to their default values.

12.4.2 Types of System Registers

Memory Size (system register 0)

Set the size of the memory area for the user program.

Hold On/Off (system registers 5–8, 10–14)

Use these system registers to specify the hold area start addresses for relays and registers. Hold areas are not cleared to 0 when the PLC is switched to PROG mode or when the power is turned off.

The memory area for timer relays and counter relays is partitioned using system register no. 5. Specify the start address for the counter relays.

Act on Error (system registers 4, 20, 23, 26)

Set the operation mode to be chosen after errors such as an operation error, a battery error, or an I/O verification error.

Time-Out (system registers 30–32, 34)

Set the waiting time before an error is output. You can also specify a constant scan time.

PLC Link (system registers 40–47, 50–55, 57)

These settings are for using link relays and link registers in MEWNET-W0 PLC link communication. Note that PLC Link is not the default setting.

High-Speed Counter, Pulse-Catch Input, Interrupt Input (system registers 400–405)

When using the high-speed counter function, pulse catch function or interrupt function, set the operation mode and the input number to be used for the function.

Time Constants (system registers 430–433)

Set a time constant for the CPU inputs. These time constants can be useful to negate the effects of noise or bouncing, e.g. for a switching device.

TOOL Port, COM Port (system registers 410–421)

Set these registers when the TOOL port and COM ports 1 and 2 ports are to be used for MEWTOCOL-COM Master/Slave connections, program controlled communication, PLC link, and modem communication. Note that the default setting is MEWTOCOL-COM Master/Slave.

12.4.3 Checking and Changing System Registers

◆Procedure

1. **Double-click "PLC" in the navigator**
2. **Double-click "System Registers"**
3. **To change a value, write the new value into the system register table**
4. **Online → Online Mode or**
5. **Online → Download Program Code and PLC Configuration**

   This downloads the project and system registers.

   To download system registers only:

6. **Online ® PLC Configuration**
7. **Select “System Registers”**
8. **Choose [Download to PLC]**

12.4.4 Table of System Registers

Memory Size

No.	Name	Default	Values
0	Sequence program area size	12/16/32 kwords[1)]	Fixed

1) Depending on PLC type (12k, 16k, or 32k type)

Hold On/Off [1)]

No.	Name	Default	Values
5	Counter start address	1008	0–1024
6	Timer/Counter hold area start address	1008	Fixed/0–1024[3)]
7	Internal relay hold area start address (in word units)	248	Fixed/0–256[3)]
8	Data register hold area start address	12000/ 32450[2)]	Fixed/0–32763[3)]
10	Link relay hold area start address for PLC Link 0 (in word units)	64	Fixed/0–64[3)]
11	Link relay hold area start address for PLC Link 1 (in word units)	128	Fixed/64–128[3)]
12	Link register hold area start address for PLC Link 0	128	Fixed/0–128[3)]
13	Link register hold area start address for PLC Link 1	256	Fixed/128–256[3)]
14	Step ladder hold/non-hold	Non-hold	Fixed or Hold/Non-hold3)

1) FP0R-T32: If the battery is empty and additional hold areas have been defined, the hold/non-hold operation becomes unstable. The data value will become indefinite. It is cleared to 0 the next time the power is turned on.

2) Depending on PLC type (16k/32k type)

3) Depending on PLC type (Fixed for C10, C14, C16, C32, variable for T32, F32)

Act on Error

No.	Name	Default	Values
4	DF-, P-function leading/falling edge detection	Holds result	Holds result/disregards result
20	Duplicate output	Enable	Fixed
23	I/O verification error	Stop	Stop/Continue
26	Operation error	Stop	Stop/Continue

Time-Out

No.	Name	Default	Values
30	Watchdog timer time-out	699.1ms	Fixed
31	Multi-frame communication time	6500.0ms	10.0–81900.0ms
32	Timeout value for the communication functions based on F145, F146, F152, F153	10000.0ms	10.0–81900.0ms
34	Constant scan time	0.0ms	0.0–600.0ms 0.0: Normal scan (non-constant)

PLC Link

No.	Name	Default	Values
46	PLC Link 0 and 1 allocation setting	Normal	Normal/Reverse
47	PLC link 0 - Highest station number in network	16	1–16
40	PLC link 0 - Link relays - Send/receive area - Number of words shared by all linked PLCs	0	0–64 words
42	PLC link 0 - Link relays - Send area - Start sending from this word address	0	0–63
43	PLC link 0 - Link relays - Send area - Number of words to send	0	0–64 words
41	PLC link 0 - Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0–128 words
44	PLC link 0 - Link registers - Send area - Start sending from this word address	0	0–127
45	PLC link 0 - Link registers - Send area - Number of words to send	0	0–127 words
57	PLC link 1 - Highest station number in network	16	1–16
50	PLC link 1 - Link relays - Send/receive area - Number of words shared by all linked PLCs	0	0–64 words
52	PLC link 1 - Link relays - Send area - Start sending from this word address	64	64–127
53	PLC link 1 - Link relays - Send area - Number of words to send	0	0–64 words
51	PLC link 1 - Link registers - Send/receive area - Number of words shared by all linked PLCs	0	0–128 words
54	PLC link 1 - Link registers - Send area - Start sending from this word address	128	128–255
55	PLC link 1 - Link registers - Send area - Number of words to send	0	0–127 words

High-Speed Counter, Pulse-Catch Input, Interrupt Input

No.	Name	Default	Values
400	High-speed counter: Channel 0	Unused	• Two-phase input (X0, X1) • Two-phase input (X0, X1), Reset input (X2) • Incremental input (X0) • Incremental input (X0), Reset input (X2) • Decremental input (X0) • Decremental input (X0), Reset input (X2) • Incremental input (X0), Decremental input (X1) • Incremental input (X0), Decremental input (X1), Reset input (X2) • Counter input (X0), Incremental/decremental control input (X1) • Counter input (X0), Incremental/decremental control input (X1), Reset input (X2)
400	High-speed counter: Channel 1	Unused	• Incremental input (X1) • Incremental input (X1), Reset input (X2) • Decremental input (X1) • Decremental input (X1), Reset input (X2)

No.	Name	Default	Values
400	High-speed counter: Channel 2	Unused	• Two-phase input (X3, X4) • Two-phase input (X3, X4), Reset input (X5) • Incremental input (X3) • Incremental input (X3), Reset input (X5) • Decremental input (X3) • Decremental input (X3), Reset input (X5) • Incremental input (X3), Decremental input (X4) • Incremental input (X3), Decremental input (X4), Reset input (X5) • Counter input (X3), Incremental/decremental control input (X4) • Counter input (X3), Incremental/decremental control input (X4), Reset input X5)
400	High-speed counter: Channel 3	Unused	• Incremental input (X4) • Incremental input (X4), Reset input (X5) • Decremental input (X4) • Decremental input (X4), Reset input (X5)
401	High-speed counter: Channel 4	Unused	• Two-phase input (X6, X7) • Incremental input (X6) • Decremental input (X6) • Incremental input (X6), Decremental input (X7) • Counter input (X6), Incremental/decremental control input (X7)
401	High-speed counter: Channel 5	Unused	• Incremental input (X7) • Decremental input (X7)
402	Pulse output: Channel 0 (transistor types only)	Unused	• Pulse output (Y0, Y1) • Pulse output (Y0, Y1), Home input (X4) • Pulse output (Y0, Y1), Home input (X4), Position control trigger input (X0) • PWM output (Y0)
402	Pulse output: Channel 1 (transistor types only)	Unused	• Pulse output (Y2, Y3) • Pulse output (Y2, Y3), Home input (X5) • Pulse output (Y2, Y3), Home input (X5), Position control trigger input (X1) • PWM output (Y2)
402	Pulse output: Channel 2 (transistor types only)	Unused	• Pulse output (Y4, Y5) • Pulse output (Y4, Y5), Home input (X6) • Pulse output (Y4, Y5), Home input (X6), Position control trigger input (X2) • PWM output (Y4)
402	Pulse output: Channel 3 (transistor types only)	Unused	• Pulse output (Y6, Y7) • Pulse output (Y6, Y7), Home input (X7) • Pulse output (Y6, Y7), Home input (X7), Position control trigger input (X3) • PWM output (Y6)
403	Pulse catch input: X0	Disable	Disable/Enable

No.	Name	Default	Values
403	Pulse catch input: X1	Disable	Disable/Enable
403	Pulse catch input: X2	Disable	Disable/Enable
403	Pulse catch input: X3	Disable	Disable/Enable
403	Pulse catch input: X4	Disable	Disable/Enable
403	Pulse catch input: X5	Disable	Disable/Enable
403	Pulse catch input: X6	Disable	Disable/Enable
403	Pulse catch input: X7	Disable	Disable/Enable
404/ 405	Interrupt input: X0→Interrupt 0	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X1→Interrupt 1	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X2→Interrupt 2	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X3→Interrupt 3	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X4→Interrupt 4	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X5→Interrupt 5	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X6→Interrupt 6	Unused	Rising edge/Falling edge/Rising and falling edge
404/ 405	Interrupt input: X7→Interrupt 7	Unused	Rising edge/Falling edge/Rising and falling edge

- **If the same input has been set as high-speed counter input, pulse catch input or interrupt input, the following order of precedence is effective: High-speed counter → Pulse catch → Interrupt.**
- **If reset input settings overlap for channel 0 and channel 1, the channel 1 setting takes precedence. If reset input settings overlap for channel 2 and channel 3, the channel 3 setting takes precedence.**
- **The input modes two-phase, incremental/decremental, or incremental/decremental control require a second channel. If channel 0, 2, or channel 4 has been set to one of these modes, the settings for channel 1, 3, and 5, respectively, will be invalid.**
- **The settings for pulse catch inputs and interrupt inputs can only be specified in the system registers.**

Transistor types (C16 and higher)

- **CPU outputs which have been specified as pulse output or PWM output cannot be used as normal outputs.**
- **Input numbers X4 to X7 can be used as home input of pulse output channels 0 to 3. When using the home return function, always set the home input. In this case, X4 to X7 cannot be used as high-speed counter inputs.**
- **The output numbers for the deviation counter clear signal, which can**

be used with the home return function, are fixed for each channel.
For C16: Channel 0 = Y6, channel 1 = Y7
For C32/T32/F32: Channel 0 = Y8, channel 1 = Y9, channel 2 = YA, channel 3 = YB
If used for the deviation counter clear signal, these outputs are not available as pulse outputs.

Time Constants

No.	Name	Default	Values
430	Time constant of input X0	1.0ms	0.1ms 0.5ms 1.0ms 2.0ms 4.0ms 8.0ms 16.0ms 32.0ms 64.0ms
430	Time constant of input X1		
430	Time constant of input X2		
430	Time constant of input X3		
431	Time constant of input X4		
431	Time constant of input X5		
431	Time constant of input X6		
431	Time constant of input X7		
432[1)]	Time constant of input X8		
432[1)]	Time constant of input X9		
432[1)]	Time constant of input XA		
432[1)]	Time constant of input XB		
433[1)]	Time constant of input XC		
433[1)]	Time constant of input XD		
433[1)]	Time constant of input XE		
433[1)]	Time constant of input XF		

[1)] 32k types only

TOOL Port

No.	Name	Default	Values
412	TOOL port - communication mode	MEWTOCOL-COM Slave	MEWTOCOL-COM Slave/Program controlled
410	TOOL port -station number	1	1–99
415	TOOL port - baud rate	115200 baud	115200/57600/38400/19200/9600/4800/2400 baud
413	TOOL port - sending data length	8 bits	7 bits/8 bits
413	TOOL port -sending parity check	With-Odd	None/With-Odd/With-Even
413	TOOL port - sending stop bit	1 bit	1 bit/2 bits
413	TOOL port - sending start code	No-STX	No-STX/STX
413	TOOL port - sending end code/reception done condition	CR	CR/CR+LF/ETX/None
420	TOOL port- -receive buffer starting address	0	0–12312 (16k type) 0–32762 (32k type)
421	TOOL port - receive buffer capacity	0	0-2048

No.	Name	Default	Values
412	TOOL port - modem connection	Disable	Disable/Enable

COM Port

No.	Name	Default	Values
412	COM port 1 - communication mode	MEWTOCOL-COM Master/Slave	MEWTOCOL-COM Master/Slave/Program controlled/PLC Link/Modbus RTU Master/Slave
410	COM port 1 -station number	1	1–99
415	COM port 1 - baud rate[1)]	9600 baud	115200/57600/38400/19200/9600/4800/2400 baud
413	COM port 1 - sending data length	8 bits	7 bits/8 bits
413	COM port 1 -sending parity check[1)]	With-Odd	None/With-Odd/With-Even
413	COM port 1 - sending stop bit	1 bit	1 bit/2 bits
413	COM port 1 - sending start code[1)]	No-STX	No-STX/STX
413	COM port 1 - sending end code/reception done condition[1)]	CR	CR/CR+LF/ETX/None
416	COM port 1- -receive buffer starting address	0	0–12312 (16k type) 0–32762 (32k type)
417	COM port 1 - receive buffer capacity	0	0-2048
412	COM port 1 - modem connection	Disable	Disable/Enable

[1)] For PLC Link, the communication format and baud rate settings are fixed:
Data length: 8 bits
Parity: Odd
Stop bit: 1 bit
End code: CR
Start code: No STX
Other system register settings will be ignored.

12.5 Error Codes

12.5.1 Error Codes E1 to E8

Error code	Name of error	Operation status of PLC	Description and steps to take
E1 (see note)	Syntax error	Stops	A program with a syntax error has been written. Change to PROG mode and correct the error.
E2 (see note)	Duplicated output error	Stops	Two or more operation results are output to the same relay. (This error also occurs if the same timer/counter number is being used.) Change to PROG mode and correct the error. This error is also detected during online editing. No changes will be downloaded and operation will continue.
E3	Not paired error	Stops	For instructions which must be used in a pair such as jump (JP and LBL), one instruction is either missing or in an incorrect position. Change to PROG mode and correct the error.
E4 (see note)	Parameter mismatch error	Stops	An instruction has been written which does not agree with system register settings. For example, the timer/counter number setting in a program does not agree with the timer/counter range setting. Change to PROG mode and correct the error.
E5 (see note)	Program area error	Stops	An instruction was written to the wrong program area (main program area or subprogram area) Change to PROG mode and correct the error. This error is also detected during online editing. No changes will be downloaded and operation will continue.
E6 (see note)	Compile memory full error	Stops	The program stored in the PLC is too large to compile in the program memory. Change to PROG mode and correct the error.
E7 (see note)	High-level instruction type error	Stops	In the program, high-level F and P instructions are triggered by the same operation result. (While the execution condition is TRUE, F instructions are executed in every scan. P instructions are executed only once, at the leading edge of the execution condition.) Correct the program so that the high-level instructions executed in every scan and at the leading edge are triggered separately.
E8	High-level instruction operand combination error	Stops	There is an incorrect operand in an instruction which requires a specific combination of operands (for example, the operands must all be of a certain type). Change to PROG mode and correct the error.

◆**NOTE**

In FPWIN Pro, these errors are detected by the compiler. Therefore, they are not critical.

12.5.2 Self-Diagnostic Error Codes

Error code	Name of error		Operation status of PLC	Description and steps to take
E26	User's ROM error		Stops	Probably a hardware problem. Please contact your dealer.
E27	Unit installation error		Stops	The number of installed units exceeds the limit. Turn off the power supply and check the restrictions on unit combinations.
E28	System register error		Stops	Probably an error in the system registers. Check the system register settings.
E30	Interrupt error 0		Stops	Probably a hardware problem. Please contact your dealer.
E31	Interrupt error 1		Stops	An interrupt occurred without an interrupt request. A hardware problem or error due to noise is possible. Turn off the power and check the noise conditions.
E32	Interrupt error 2		Stops	An interrupt occurred without an interrupt request. A hardware problem or error due to noise is possible. Turn off the power and check the noise conditions.
				There is no interrupt program for an interrupt which occurred. Check the number of the interrupt program and change it to agree with the interrupt request.
E34	I/O status error		Stops	A faulty unit is installed. Replace the unit with a new one.
E42	I/O unit verify error		Selectable	The connection condition of an I/O unit has changed compared to that at the time of power-up. Check the error using sys_wVerifyErrorUnit_0_15 and locate the faulty I/O unit. Set the operation status using system register 23 to continue operation.
E45	Operation error		Selectable	Operation became impossible when a high-level instruction was executed. The causes of calculation errors vary depending on the instruction. Set the operation status using system register 23 to continue operation.
E100–E299	Self-diagnostic error set by F148_ERR	E100–E199	Stops	The self-diagnostic error specified by the F148_ERR instruction occurred. Take steps to clear the error condition according to the specification you chose.
		E200–E299	Continues	

12.5.3 MEWTOCOL-COM Error Codes

Error code	Name	Description
!21	NACK error	Link system error
!22	WACK error	
!23	Unit no. overlap	
!24	Transmission format error	
!25	Link unit hardware error	
!26	Unit no. setting error	
!27	No support error	
!28	No response error	

Error code	Name	Description
!29	Buffer closed error	
!30	Time-out error	
!32	Transmission impossible error	
!33	Communication stop	
!36	No destination error	
!38	Other communication error	
!40	BCC error	A transfer error occurred in the data received.
!41	Format error	A formatting error in the command received was detected.
!42	No support error	A non-supported command was received.
!43	Multiple frames procedure error	A different command was received when processing multiple frames.
!50	Link setting error	A non-existing route number was specified. Verify the route number by designating the transmission station.
!51	Transmission time-out error	Transmission to another device is not possible because the transmission buffer is full.
!52	Transmit disable error	Transmission processing to another device is not possible (link unit runaway, etc.).
!53	Busy error	Processing of command received is not possible because of multiple frame processing or because command being processed is congested.
!60	Parameter error	Content of specified parameter does not exist or cannot be used.
!61	Data error	There was a mistake in the contact, data area, data number designation, size designation, range, or format designation.
!62	Registration over error	Operation was done when number of registrations was exceeded or when there was no registration.
!63	PC mode error	PC command that cannot be processed was executed during RUN mode.
!64	External memory error	An abnormality occurred when loading RAM to ROM/IC memory card. There may be a problem with the ROM or IC memory card. When loading, the specified contents exceeded the capacity. Write error occurs. • ROM or IC memory card is not installed. • ROM or IC memory card does not conform to specifications
!65	Protect error	A program or system register write operation was executed when the protect mode (password setting or DIP switch, etc.) or ROM operation mode was being used.
!66	Address error	There was an error in the code format of the address data. Also, when exceeded or insufficient address data, there was a mistake in the range designation.
!67	No program error and no data error	Cannot be read because there is no program in the program area or the memory contains an error. Or, reading of non-registered data was attempted.
!68	Rewrite during RUN error	When inputting with programming tool software, editing of an instruction (ED, SUB, RET, INT, IRET, SSTP, and STPE) that cannot perform a rewrite during RUN is being attempted. Nothing is written to the CPU.
!70	SIM over error	Program area was exceeded during a program write process.
!71	Exclusive access control error	A command that cannot be processed was executed at the same time as a command being processed.

12.6 MEWTOCOL-COM Communication Commands

Command name	Code	Description
Read contact area	RC (RCS) (RCP) (RCC)	Reads the on and off status of contacts. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Write contact area	WC (WCS) (WCP) (WCC)	Turns contacts on and off. - Specifies only one point. - Specifies multiple contacts. - Specifies a range in word units.
Read data area	RD	Reads the contents of a data area.
Write data area	WD	Writes data to a data area.
Read timer/counter set value area	RS	Reads the value set for a timer/counter.
Write timer/counter set value area	WS	Writes a timer/counter setting value.
Read timer/counter elapsed value area	RK	Reads the timer/counter elapsed value.
Write timer/counter elapsed value area	WK	Writes the timer/counter elapsed value.
Register or Reset contacts monitored	MC	Registers the contact to be monitored.
Register or Reset data monitored	MD	Registers the data to be monitored.
Monitoring start	MG	Monitors a registered contact or data using MD and MC.
Preset contact area (fill command)	SC	Embeds the area of a specified range in a 16-point on and off pattern.
Preset data area (fill command)	SD	Writes the same contents to the data area of a specified range.
Read system register	RR	Reads the contents of a system register.
Write system register	WR	Specifies the contents of a system register.
Read the status of PLC	RT	Reads the specifications of the PLC and error codes if an error occurs.
Remote control	RM	Switches the operation mode of the PLC.
Abort	AB	Aborts communication.

12.7 Data Types

FPWIN Pro provides elementary and user defined data types.

Elementary data types

Data Type	Abbreviation	Description	Range
BOOL	BOOL	Boolean (1 bit)	0 (FALSE) or 1 (TRUE)
INTEGER	INT	Integer (16 bits)	-32768–32767
UNSIGNED INTEGER	UINT	Unsigned integer (16 bits)	0–65535
DOUBLE INTEGER	DINT	Double integer (32 bits)	-2147483648–2147483647
UNSIGNED DOUBLE INTEGER	UDINT	Unsigned double integer (32 bits)	0–4294967295
REAL	REAL	Real number (32 bits)	$-3.402823E^{38}$– $-1.17549410E^{-38}$, 0, $+1.17549410E^{-38}$– $+3.402823E^{38}$
WORD	WORD	Bit string of length 16 (16 bits)	16#0–16#FFFF
DOUBLE WORD	DWORD	Bit string of length 32 (32 bits)	16#0–16#FFFFFFFF
TIME[1)]	TIME	Duration (32 bits)	T#0s– T#248d3h13m56s470ms
DATE AND TIME[1)]	DT	Date and time of day (32 bits)	DT#2001-01-01-00:00:00– DT#2099-12-31-23:59:59
TIME OF DAY[1)]	TOD	Time of day (only) (32 bits)	TOD#00:00:00– TOD#23:59:59
DATE[1)]	DATE	Date (only) (32 bits)	DT#2001-01-01– DT#2099-12-31
STRING[1)]	STRING	Variable-length character string	Depending on PLC type
ARRAY	ARRAY	Number of similar data elements	Depending on PLC type

A data type has to be assigned to each variable.

[1)]Internal representation of time and string data types

TIME	In 10ms units
DATE AND TIME	Seconds after DT#2001-01-01-00:00:00
TIME OF DAY	Seconds after TOD#00:00:00
DATE	Seconds after DT#2001-01-01
STRING	Header with maximum and current number of characters followed by the characters

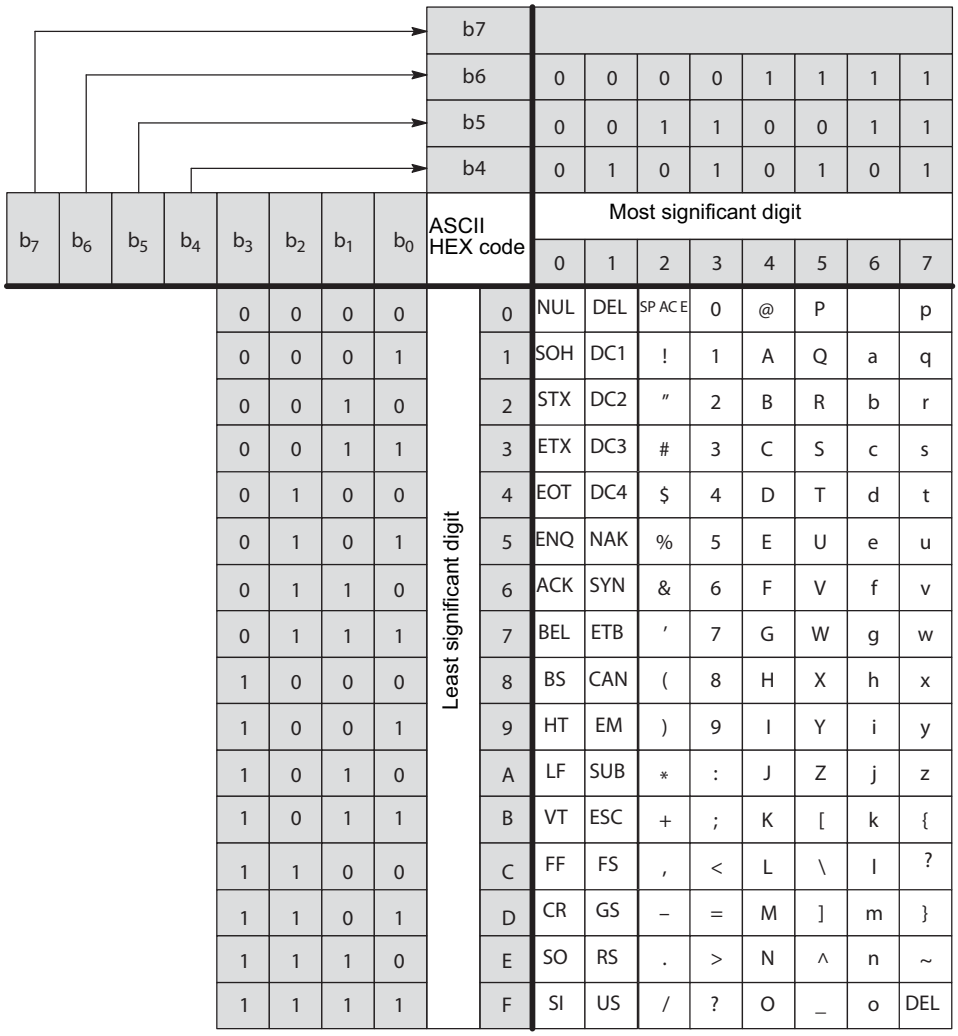
User defined data types

We differentiate between **array** and **D**ata **U**nit **T**ypes (DUT). An array consists of several elementary data types which are all of the same type. A DUT consists of several elementary data types of different types. Each represents a new data type.

12.8 Hexadecimal/Binary/BCD

Decimal	Hexadecimal	Binary data	BCD data (Binary Coded Decimal)
0	0000	0000 0000 0000 0000	0000 0000 0000 0000
1	0001	0000 0000 0000 0001	0000 0000 0000 0001
2	0002	0000 0000 0000 0010	0000 0000 0000 0010
3	0003	0000 0000 0000 0011	0000 0000 0000 0011
4	0004	0000 0000 0000 0100	0000 0000 0000 0100
5	0005	0000 0000 0000 0101	0000 0000 0000 0101
6	0006	0000 0000 0000 0110	0000 0000 0000 0110
7	0007	0000 0000 0000 0111	0000 0000 0000 0111
8	0008	0000 0000 0000 1000	0000 0000 0000 1000
9	0009	0000 0000 0000 1001	0000 0000 0000 1001
10	000A	0000 0000 0000 1010	0000 0000 0001 0000
11	000B	0000 0000 0000 1011	0000 0000 0001 0001
12	000C	0000 0000 0000 1100	0000 0000 0001 0010
13	000D	0000 0000 0000 1101	0000 0000 0001 0011
14	000E	0000 0000 0000 1110	0000 0000 0001 0100
15	000F	0000 0000 0000 1111	0000 0000 0001 0101
16	0010	0000 0000 0001 0000	0000 0000 0001 0110
17	0011	0000 0000 0001 0001	0000 0000 0001 0111
18	0012	0000 0000 0001 0010	0000 0000 0001 1000
19	0013	0000 0000 0001 0011	0000 0000 0001 1001
20	0014	0000 0000 0001 0100	0000 0000 0010 0000
21	0015	0000 0000 0001 0101	0000 0000 0010 0001
22	0016	0000 0000 0001 0110	0000 0000 0010 0010
23	0017	0000 0000 0001 0111	0000 0000 0010 0011
24	0018	0000 0000 0001 1000	0000 0000 0010 0100
25	0019	0000 0000 0001 1001	0000 0000 0010 0101
26	001A	0000 0000 0001 1010	0000 0000 0010 0110
27	001B	0000 0000 0001 1011	0000 0000 0010 0111
28	001C	0000 0000 0001 1100	0000 0000 0010 1000
29	001D	0000 0000 0001 1101	0000 0000 0010 1001
30	001E	0000 0000 0001 1110	0000 0000 0011 0000
31	001F	0000 0000 0001 1111	0000 0000 0011 0001
.	.	.	.
.	.	.	.
.	.	.	.
63	**003F**	0000 0000 0011 1111	0000 0000 0110 0011
.	.	.	.
.	.	.	.
.	.	.	.
255	**00FF**	0000 0000 1111 1111	0000 0010 0101 0101
.	.	.	.
.	.	.	.
.	.	.	.
9999	**270F**	0010 0111 0000 1111	1001 1001 1001 1001

12.9 ASCII Codes



b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	ASCII HEX code		b7							
										b6	0	0	0	0	1	1	1	1
										b5	0	0	1	1	0	0	1	1
										b4	0	1	0	1	0	1	0	1
											Most significant digit							
											0	1	2	3	4	5	6	7
				0	0	0	0	Least significant digit	0		NUL	DEL	SP AC E	0	@	P		p
				0	0	0	1		1		SOH	DC1	!	1	A	Q	a	q
				0	0	1	0		2		STX	DC2	″	2	B	R	b	r
				0	0	1	1		3		ETX	DC3	#	3	C	S	c	s
				0	1	0	0		4		EOT	DC4	$	4	D	T	d	t
				0	1	0	1		5		ENQ	NAK	%	5	E	U	e	u
				0	1	1	0		6		ACK	SYN	&	6	F	V	f	v
				0	1	1	1		7		BEL	ETB	′	7	G	W	g	w
				1	0	0	0		8		BS	CAN	(	8	H	X	h	x
				1	0	0	1		9		HT	EM	)	9	I	Y	i	y
				1	0	1	0		A		LF	SUB	*	:	J	Z	j	z
				1	0	1	1		B		VT	ESC	+	;	K	[	k	{
				1	1	0	0		C		FF	FS	,	<	L	\	l	?
				1	1	0	1		D		CR	GS	–	=	M	]	m	}
				1	1	1	0		E		SO	RS	.	>	N	^	n	~
				1	1	1	1		F		SI	US	/	?	O	_	o	DEL

Index

1

1:1
- Communication....85, 126
- Communication with a Computer....105
- Communication with GT Series Programmable Displays....106
- Communication with Micro-Imagechecker....127

1:N
- Slave Communication....104
- Communication....133
- Slave Communication....107

A

Accessories....11

All LEDs are OFF....218

Appendix....223

ASCII Codes....255

B

Backup and Clock/Calendar Functions....31

Backup Function....32

Battery....33

BEFORE BEGINNING....i

C

C10 CPU....27

C10/C14 CPU (Terminal Block)....231

C14 CPU....28

C16 CPU....29

C16 CPU (MIL Connector)....233

C32 CPU....30

C32 CPU (MIL Connector)....235

Calendar function....33

Changing Communication Mode in RUN Mode....96

Checking and Changing System Registers....242

Clock function....33

Clock/Calendar Function....33

COM port....18, 82, 86, 93

COM Port....87

Command and Response Format....100

Commands....102

Comment memory....6

Communication....81

Communication Modes....82

Communication Parameters....94

Communication Specifications....93, 227

Communication with the Programming Tool....91

Connecting FP0 Expansion Units....63

Connector....18, 41

Count Input Modes....166

CPU....8, 51

CPU Types....17

Current Consumption....228

D

Data Format....120

Data Types....253

Diagnosing Output Malfunction....219

Dimensions....231

DIN rail attachment lever....18, 41

Download Protection....206

E

Error Codes....249

Error Codes E1 to E8....249

Error Detection Time for Transmission Assurance Relays....150

ERROR/ALARM LED is Flashing....216

ERROR/ALARM LED is ON 217
Example for PLC link 0 139
Example for PLC link 1 140
Expansion 39
Expansion hook 18, 41
Expansion Method 40

F

F166_HighSpeedCounter_Set, Target Value Match ON 173
F166_PulseOutput_Set, Target Value Match ON 189
F167_HighSpeedCounter_Reset, Target Value Match OFF 174
F167_PulseOutput_Reset, Target Value Match OFF 190
F171_PulseOutput_Jog_Positioning, JOG Operation and Positioning 192
F171_PulseOutput_Trapezoidal, Trapezoidal Control 191
F172_PulseOutput_Jog, JOG Operation 194
F174_PulseOutput_DataTable, Data Table Control 195
F175_PulseOutput_Linear, Linear Interpolation 196
F177_PulseOutput_Home, Home Return 196
F178_HighSpeedCounter_Measure, Input Pulse Measurement 175
Features 6
Flag Operation 121
Flat Type Mounting Plate 60
FP Memory Loader 13, 205
FP Series Link Units 10
FP0 Compatibility Mode 103, 112
FP0 Expansion Units 52
FP0 I/O Expansion Units 9
FP0 Intelligent Units 9
FP0 Program Compatibility 14
FPWIN Pro .....iv, 6, 13, 14, 82
F-ROM 210
F-ROM Backup (P13_EPWT) 210
Function Specifications and Restrictions 160

G

General 50
General Specifications 224
Grounding 68

H

Hexadecimal/Binary/BCD 254
High-Speed Counter and Pulse Output 157
High-speed counter control code 170
High-Speed Counter Function 160, 166

I

I/O Allocation 49, 168, 182, 229
I/O Link Unit 236
IEC 6
Important Symbols ii
Input and Output Wiring 70
Input connector 18, 41
Input pulse measurement 175
Input Specifications, CPU 22
Input Specifications, Expansion Units 43
Input Time Constants 212
Input Wiring 70
Installation 56
Installation and Wiring 55
Installation Environment and Space 56
Installing the USB Driver 89

Instructions and System Variables169, 183
Inverter, double speed 177
Inverter, single speed 176

L

LED 18
Input status 18, 41
Operation status 18
Output status 18, 41
LED Display for Status Condition 214
Link Area Allocation 138

M

Maximum Counting Speed and Output Frequency 163
Memory Area for Clock/Calendar Function 33
MEWTOCOL-COM 97
MEWTOCOL-COM Communication Commands 252
MEWTOCOL-COM Error Codes 250
MEWTOCOL-COM Master/Slave 82
Minimum Input Pulse Width 168
Modbus RTU Communication 152
Modbus RTU Master/Slave 85
Monitoring 145

O

Operating environment 2
Operation mode selector 18
Operation on Error 215
Operation Outline for MEWTOCOL-COM Slave 99
Other Functions 209
Output connector 18, 41
Output Specifications, CPU 24
Output Specifications, Expansion Units..44
Output Wiring 73
Overview 5, 158

P

Partial Use of Link Areas 142
Parts and Functions, CPU 18
Parts and Functions, Expansion Units....41
Password protection 204
Password Protection Error Message 220
Performance Specifications 225
PLC Link 84, 135
PLC Link 0 and 1 Allocation Setting 144
PLC Link Response Time 146
PLC Protection (Password Protection) .204
Ports
Names and Principle Applications 86
Positioning Operations with a Double-Speed Inverter 177
Positioning Operations with a Single-Speed Inverter 176
POU iv
Power supply 2, 41
Power Supply Unit 10, 236
Precautions for Allocating Link Areas ...143
Precautions When Setting System Registers 241
PROG Mode does not Change to RUN 221
Program Controlled Communication 83, 110
Programming Conventions iv
Programming Tools 13
Protective Circuit for Capacitive Loads...74
Protective Circuit for Inductive Loads 73
Pulse Output Function 161, 179
Pulse Output Methods and Position Control Modes 179
PWM Output Function 162, 198

R

Receiving Data from External Devices .117
Record of Changes 262
Reducing Transmission Cycle Times ...149
Reinstalling the USB Driver 91
Relays and Memory Areas for FP0R 238
Restrictions on Unit Combinations.....12

S

Safety Instructions for Wiring 64
Safety Measures 1, 2
Sample Program for 30-Second Compensation 35
Sample Program for Fixed Schedule and Automatic Start.....35
Sample Program for Master Communication 108, 155
Sample Programs 175
Sampling Trace 211
Security Function Types 202
Security Functions 201
Security Settings in FPWIN Pro.....203
Self-Diagnostic Error Codes 250
Sending Data to External Devices 113
Setting Communication Parameters 103, 111, 134, 137, 155
Setting System Registers in PROG Mode 94
Setting the Highest Station Number for a PLC Link.....144
Settings for Clock/Calendar Function 34
Slim Type Mounting Plate.....59
Specifications.....224
Start Code
No-STX
CR.....123
STX
ETX.....124
System Registers.....241
System variables.....169

T

Table of System Registers.....243
Terminal Layout 27, 46
Terminology in FPWIN Pro and FPWIN GR 82
TOOL port 18, 82, 86, 93
TOOL Port.....86
Troubleshooting 213
Types of System Registers.....241

U

Unit Types.....8
Upload Protection 203, 205
USB.....6, 18, 82, 86, 93
System requirements.....87
USB driver installation 89
USB driver reinstallation 91
USB Port 87
Using DIN Rails.....58, 237
Using Optional Mounting Plates 59

W

Wiring the COM Port.....78
Wiring the MIL Connector 75
Wiring the Power Supply 66
Wiring the Terminal Block.....76
Writing and Reading the Elapsed Value for the High-Speed Counter 173
Writing and Reading the Elapsed Value of the Pulse Output 188
Writing the High-Speed Counter Control Code.....170
Writing the Pulse Output Control Code 185

Record of Changes

Manual No.	Date	Description of changes
ARCT1F475E	May 2009	First edition
ACGM0475V1EN	August 2010	European edition • addition of FPWIN Pro examples and procedures
ACGM0475V1.1EN	February 2011	Correction of errors

North America | **Europe** | **Asia Pacific** | **China** | **Japan**

Panasonic Electric Works Global Sales Companies

Europe

▶ **Headquarters**	**Panasonic Electric Works Europe AG**	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-111, www.panasonic-electric-works.com
▶ **Austria**	**Panasonic Electric Works Austria GmbH**	Rep. of PEWDE, Josef Madersperger Str. 2, 2362 Biedermannsdorf, Tel. +43 (0) 2236-26846, Fax +43 (0) 2236-46133, www.panasonic-electric-works.at
	PEW Electronic Materials Europe GmbH	Ennshafenstraße 30, 4470 Enns, Tel. +43 (0) 7223 883, Fax +43 (0) 7223 88333, www.panasonic-electronic-materials.com
▶ **Benelux**	**Panasonic Electric Works Sales Western Europe B.V.**	De Rijn 4, (Postbus 211), 5684 PJ Best, (5680 AE Best), Netherlands, Tel. +31 (0) 499 372727, Fax +31 (0) 499 372185, www.panasonic-electric-works.nl
▶ **Czech Republic**	**Panasonic Electric Works Czech s.r.o.**	Prumtyslová 1, 34815 Planá, Tel. (+420-)374799990, Fax (+420-)374799999, www.panasonic-electric-works.cz
▶ **France**	**Panasonic Electric Works Sales Western Europe B.V.**	Succursale française, 10, rue des petits ruisseaux, 91371 Verrières le Buisson, Tél. +33 (0) 1 6013 5757, Fax +33 (0) 1 6013 5758, www.panasonic-electric-works.fr
▶ **Germany**	**Panasonic Electric Works Europe AG**	Rudolf-Diesel-Ring 2, 83607 Holzkirchen, Tel. +49 (0) 8024 648-0, Fax +49 (0) 8024 648-111 www.panasonic-electric-works.de
▶ **Hungary**	**Panasonic Electric Works Europe AG**	Magyarországi Közvetlen Kereskedelmi Képviselet, 1117 Budapest, Neumann János u. 1., Tel. +36(0)1482 9258, Fax +36 (0) 1482 9259, www.panasonic-electric-works.hu
▶ **Ireland**	**Panasonic Electric Works UK Ltd.**	Dublin, Tel. +353 (0) 14600969, Fax +353 (0) 14601131, www.panasonic-electric-works.co.uk
▶ **Italy**	**Panasonic Electric Works Italia s.r.l.**	Via del Commercio 3-5 (Z.I. Ferlina), 37012 Bussolengo (VR), Tel. +39 (0) 456752711, Fax +39 (0) 456700444, www.panasonic-electric-works.it
▶ **Nordic Countries**	**Panasonic Electric Works Nordic AB**	Sjöängsvägen 10, 19272 Sollentuna, Sweden, Tel. +46 859476680, Fax +46 859476690, www.panasonic-electric-works.se
	PEW Fire & Security Technology Europe AB	Jungmansgatan 12, 21119 Malmö, Tel. +46 40697-7000, Fax +46 40697-7099, www.panasonic-fire-security.com
▶ **Poland**	**Panasonic Electric Works Polska sp. z o.o.**	Al. Krakowska 4/6, 02-284 Warszawa, Tel. +48 (0) 22 338-11-33, Fax +48 (0) 22 338-12-00, www.panasonic-electric-works.pl
▶ **Portugal**	**Panasonic Electric Works España S.A.**	Portuguese Branch Office, Avda Adelino Amaro da Costa 728 R/C J, 2750-277 Cascais, Tel. +351 214812520, Fax +351 214812529
▶ **Spain**	**Panasonic Electric Works España S.A.**	Barajas Park, San Severo 20, 28042 Madrid, Tel. +34 913293875, Fax +34 913292976, www.panasonic-electric-works.es
▶ **Switzerland**	**Panasonic Electric Works Schweiz AG**	Grundstrasse 8, 6343 Rotkreuz, Tel. +41 (0) 417997050, Fax +41 (0) 417997055, www.panasonic-electric-works.ch
▶ **United Kingdom**	**Panasonic Electric Works UK Ltd.**	Sunrise Parkway, Linford Wood, Milton Keynes, MK14 6LF, Tel. +44(0) 1908 231555, +44(0) 1908 231599, www.panasonic-electric-works.co.uk

North & South America

▶ **USA**	**PEW Corporation of America**	629 Central Avenue, New Providence, N.J. 07974, Tel. +1-908-464-3550, Fax +1-908-464-8513, www.pewa.panasonic.com

Asia Pacific / China / Japan

▶ **China**	**Panasonic Electric Works (China) Co., Ltd.**	Level 2, Tower W3, The Tower Oriental Plaza, No. 2, East Chang An Ave., Dong Cheng District, Beijing 100738, Tel. (010) 5925-5988, Fax (010) 5925-5973, www.pewc.panasonic.cn
▶ **Hong Kong**	**Panasonic Electric Works (Hong Kong) Co., Ltd.**	RM1205-9, 12/F, Tower 2, The Gateway, 25 Canton Road, Tsimshatsui, Kowloon, Hong Kong, Tel. (8520) 2956-3118, Fax (0852) 2956-0398
▶ **Japan**	**Panasonic Electric Works Co., Ltd.**	1048 Kadoma, Kadoma-shi, Osaka 571-8686, Japan, Tel. (06)-6908-1050, Fax (06)-6908-5781 http://panasonic-electric-works.net
▶ **Singapore**	**Panasonic Electric Works Asia Pacific Pte. Ltd.**	101 Thomson Road, #25-03/05, United Square, Singapore 307591, Tel. (06255)-5473, Fax (06253)-5689

Copyright © 2011. All rights reserved. Specifications are subject to change without notice. Printed in Europe.

ACGM0475V1.1EN 02/2011